Ph. D. Thesis Presentation

딥러닝을 이용한 2D 사각 유한요소 개발
Deep Learned 2D Quadrilateral Finite Elements

2020.11.09

Jaeho Jung

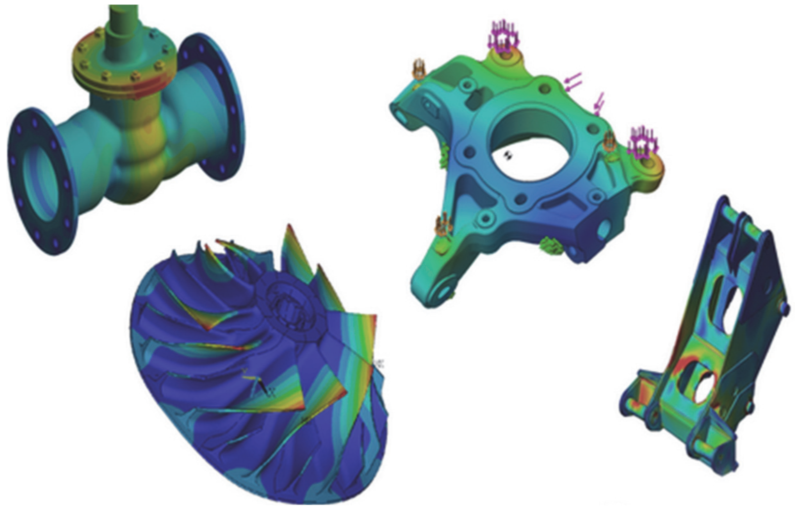
KAIST Department of Mechanical Engineering

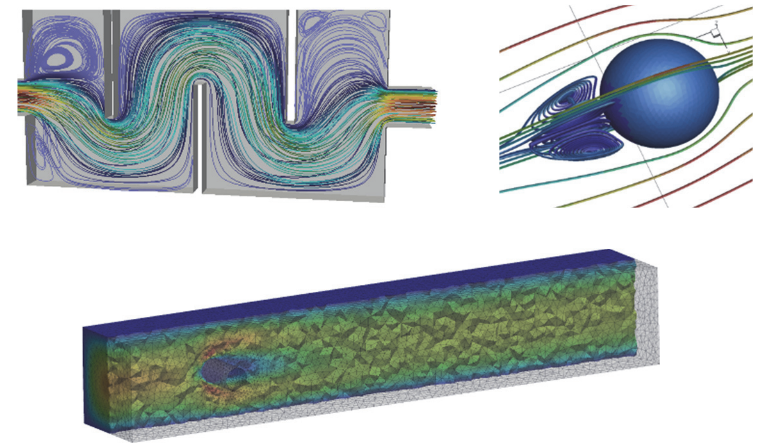
CMSS Computational Mechanics and Structural Systems

CONTENTS

I Introduction

FEM fields





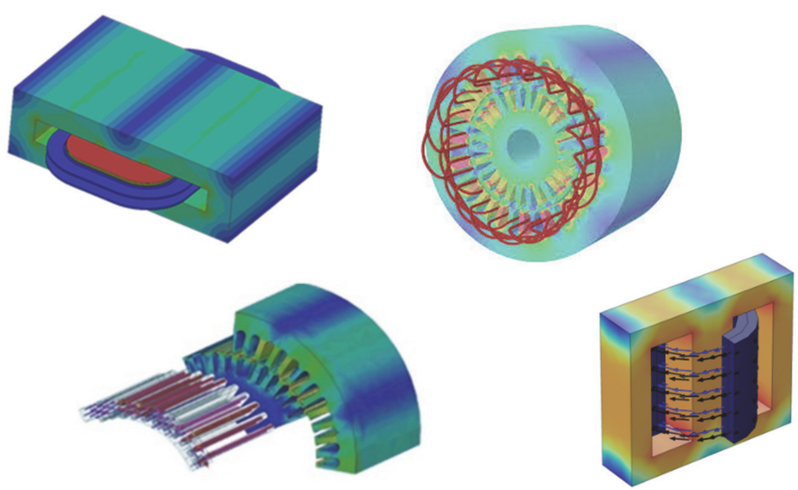
Structural analysis

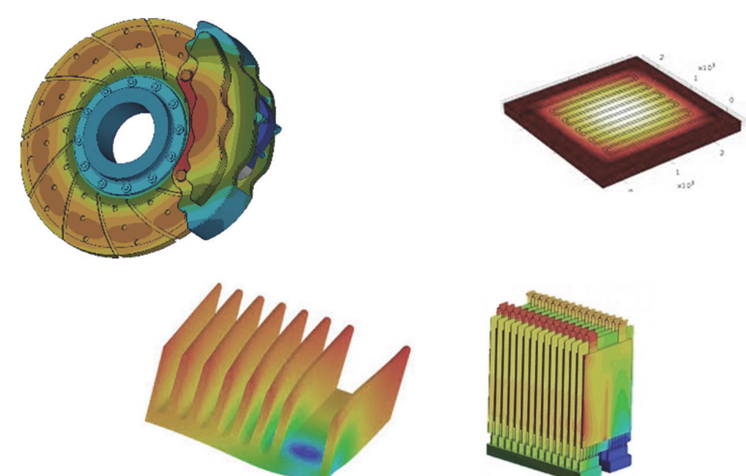
Flow simulation

Coupling

Electromagnetic analysis

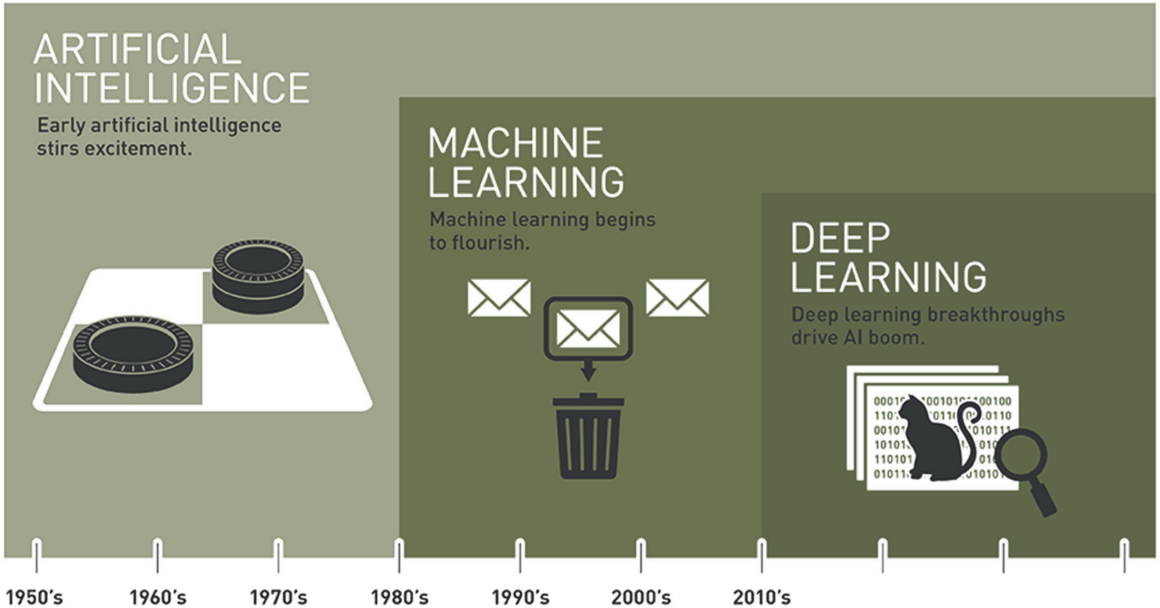
Thermal analysis

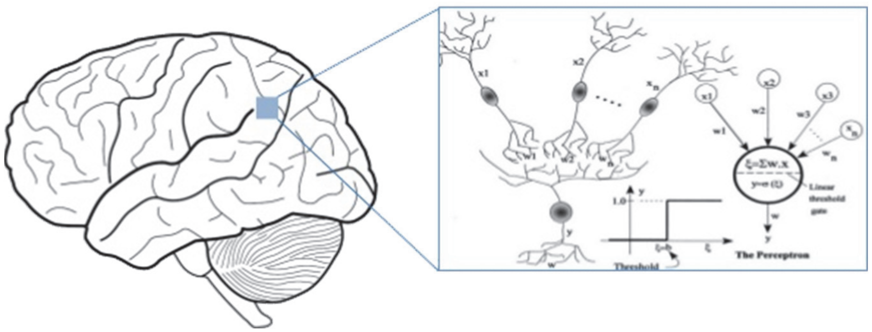


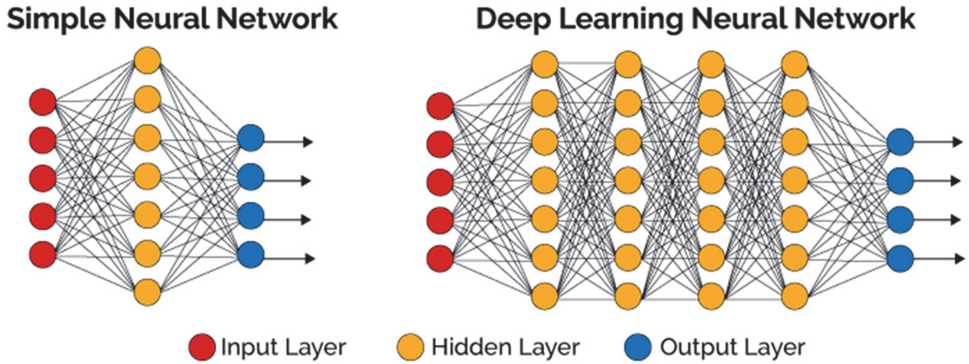


Deep Learning

- Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning from data





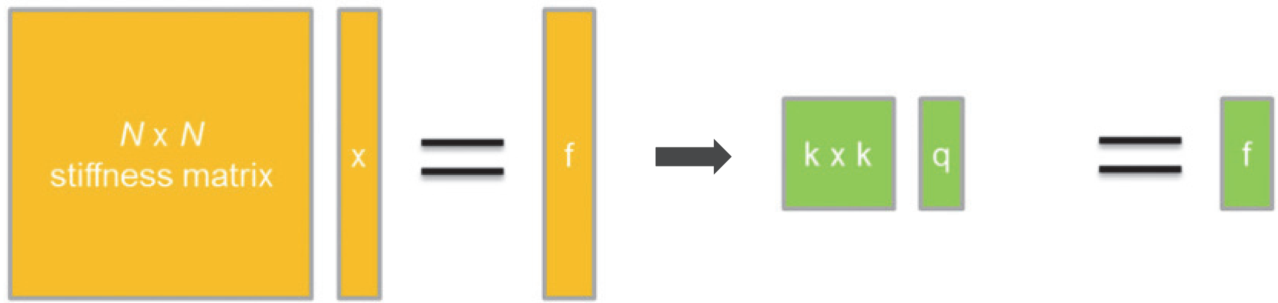


Related Work

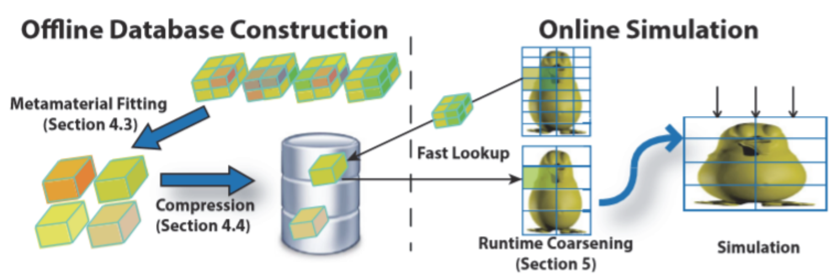
- ❖ Approaches for applying Deep Learning to numerical analysis
  - J. Sirignano and K. Spiliopoulos (2018), M. Raissi et al(2018-2019)
    - Solving PDE (learning must be performed for each PDE)
  - Z.J. Zhang et al (2015), J. Ling et al (2016), X. Guo et al (2016), J. Tompson et al (2017), O. Hennigh (2017).
    - Flow approximation (not suitable if not trained)
    - Turbulence modeling (general approach, but narrow)
  - L. Liang et al (2018), A. Chamekh et al(2009)
    - FEM surrogate model (not applicable when the domain is changed)
  - M. Amir (2020), T. Kirchdoerfer and M. Oritz (2016), Y.M. Hashash et al (2004)
    - FEM Constitutive models
  - I. Brevis et al (2020, arXiv)
    - FEM weight function parameter tuning using Petrov-Galerkin method (domain dependency problem)
  - A. Oishi and G. Yagawa (2017)
    - Improvement of numerical integration error (general approach, but slight improvement)

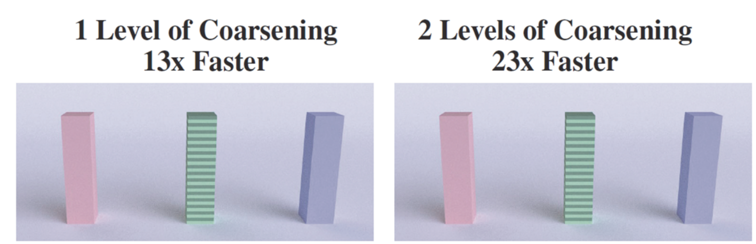
Related Work

❖ Model reduction method



- D. Chen (2015), L. Kharevych (2009), Bickel et al (2009)
  - Adapting the constitutive model



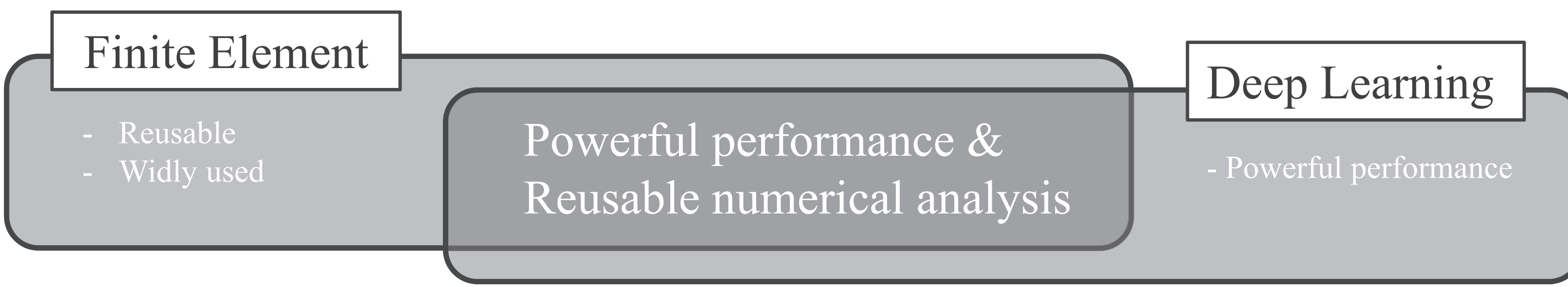


- Nesme et al (2009), Barbic and James (2005), Krysl et al. (2001), Shabana (1991)
  - Higher order shape function

Motivation and Objectives

- The most important thing of applying Deep Learning to numerical analysis
  - Generality

- Motivation
  - Developing finite elements using Deep Learning might solve the generality
    - FE feature (suitable to apply Deep Learning)
      - Geometrically bounded shape (1D Line, 2D triangle, 3D hexahedron..)
      - The results of analysis are only determined by physical value inside the element



Comments on Proposal

- Prof. Phill-Seung Lee
  - Present the applicability and limitation of proposed method. (Title, Closure (topic1), Appendix B-C)
  - Consider the solution of the limitation. (Appendix C)

- Prof. Ik Jin. Lee
  - Modify title. (Title)
  - Present the applicability and limitation of proposed method. (Title, Closure (topic1), Appendix B-C)
  - Present the differences in results according to data generation. (Appendix A, D, F)

- Prof. Seunghwa Ryu
  - Check the required training data to increase DOF. (Appendix E)
  - Consider the problem to use the proposed method to 3D element or non-linear problem. (Appendix E, Future work)
  - Consider the problem when the developed element is used to application out of the trained space. (Appendix C)

Comments on Proposal

- Prof. In Gwun Jang
  - Check whether the proposed method provides sufficient data set for elements with high DOF. (Appendix E)
  - Check whether the error can be large in a certain situation. (P22 Error distribution in the test data)
  - Suggest another method for linear element. (Topic 2)

- Ph.D Yonggyun Yu
  - More paper search for data driven FEM. (Related Work)
  - Consider cosine annealing, various activation function. (Appendix G)
  - When extending the proposed method to 3D elements.
    - Consider various sampling method. (Appendix D)
    - Robust modeling study using bootstrapping and ensemble method. (Future work)
    - Consider various optimization method. (Appendix H)

Ⅱ Research topics

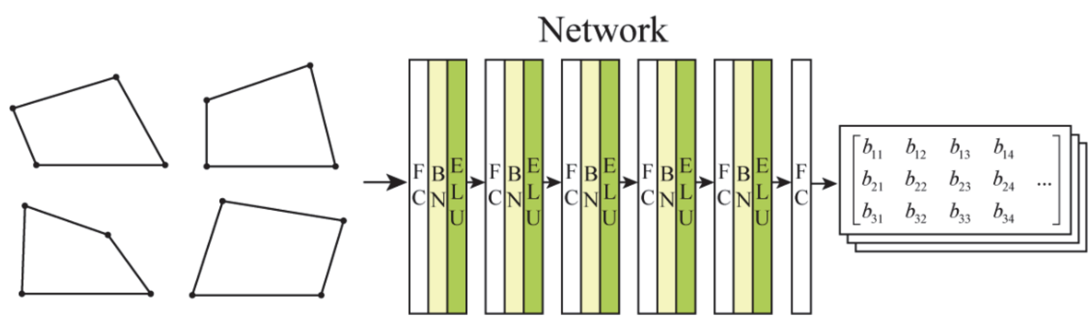
Topic 1. Deep Learned Finite Elements

Overview of DLFE

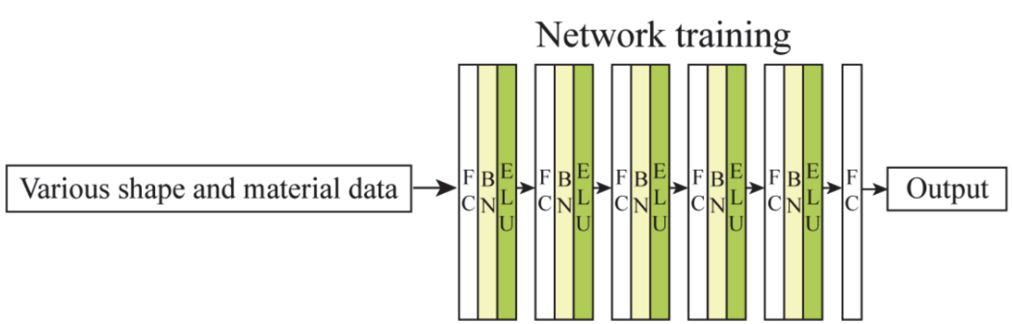
❖ Introducing the Deep learned finite elements

- Objectives
  - Making more suitable strain displacement matrix (**B**) for every shape and material without any rule teaching using deep learning.

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u} \quad \text{with} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \gamma_{xy} \end{bmatrix}^{\mathrm{T}}, \quad \mathbf{u} = \begin{bmatrix} u_1 & \dots & u_8 & v_1 & \dots & v_8 \end{bmatrix}^{\mathrm{T}}.$$



- How to make



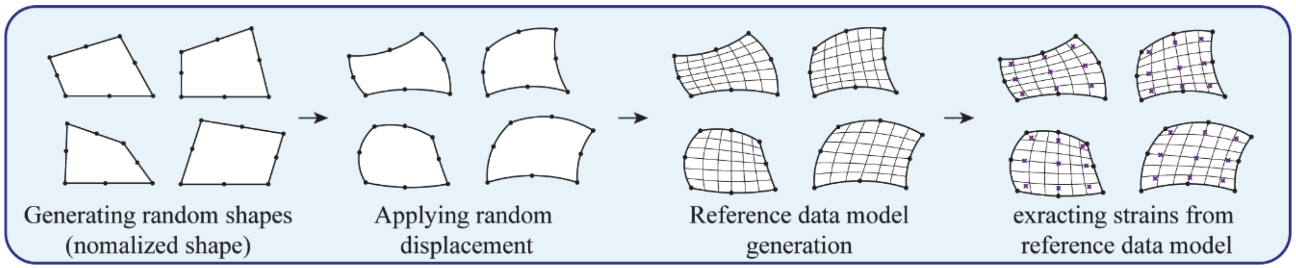
Cost function

$$C(\boldsymbol{\theta}) = \left| \frac{\mathrm{Output}(\boldsymbol{\theta})\mathbf{u} - \boldsymbol{\varepsilon}}{\boldsymbol{\varepsilon}} \right|$$

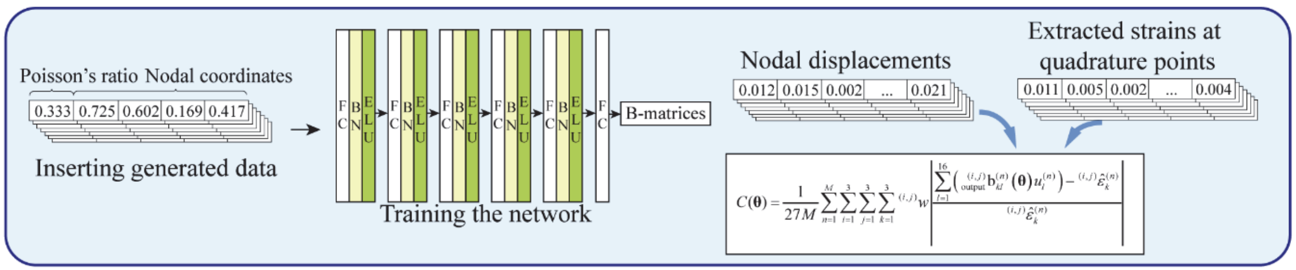
Overview of DLFE

❖ Introducing the procedure based on 8-node quadrilateral element

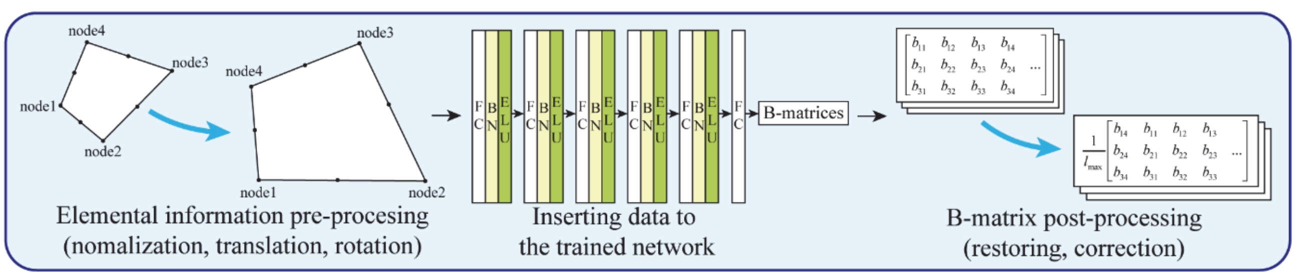
Step 1. Data generation



Step 2. Network Training



Step 3. B-matrix generation



Step 4. Elemental K-matrix generation

Generating elemental K-matrix

$$\mathbf{K} = t\sum_{i=1}^{p}\sum_{j=1}^{p} {}^{(i,j)}w^{(i,j)}\mathbf{B}^{\mathrm{T}}\mathbf{C}^{(i,j)}\mathbf{B}^{(i,j)}J$$

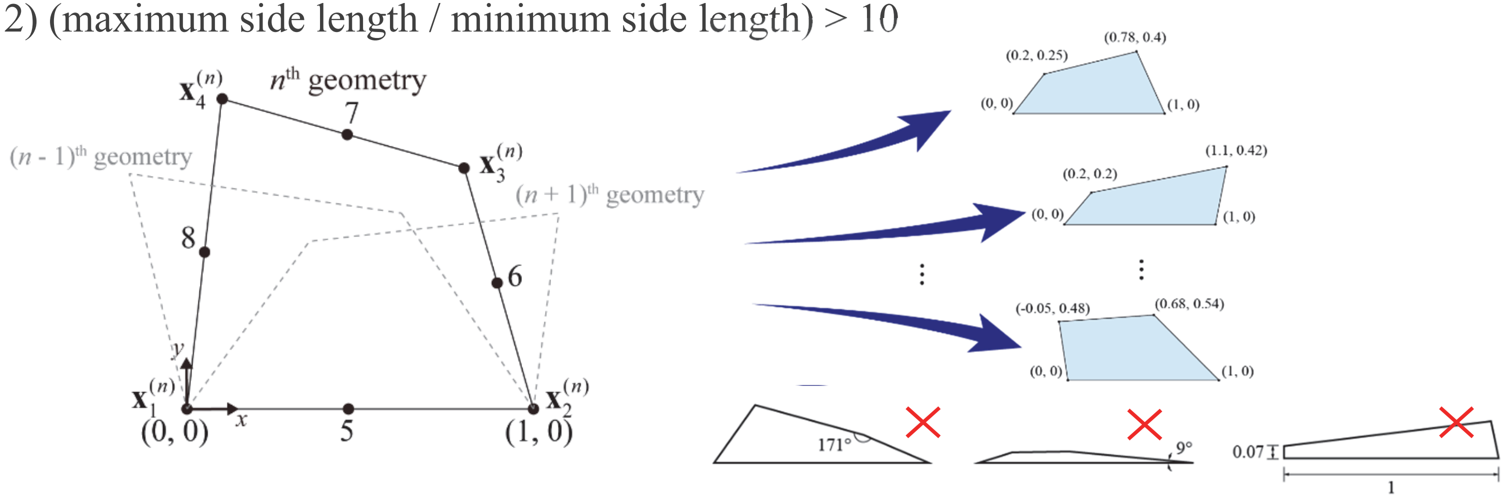
Data generation

❖ Generating strain data according to displacements, geometries, and Poisson's ratios

- Assumption & Limitation
  - the geometry of 8-node finite elements is limited to a quadrilateral whose mid-side node (nodes 5–8) are placed at the center of the adjacent corner nodes (nodes 1–4).
- Random geometry
  - The n^{th} normalized random geometry is generated as belows to do efficient learning using geometric similarity.
    - x_1, x_2 are fixed at (0, 0) and (1, 0) respectively.
    - x_3, x_4 are randomly positioned under the constraints that the side length should be less than 1
  - Severely distorted geometries are excluded

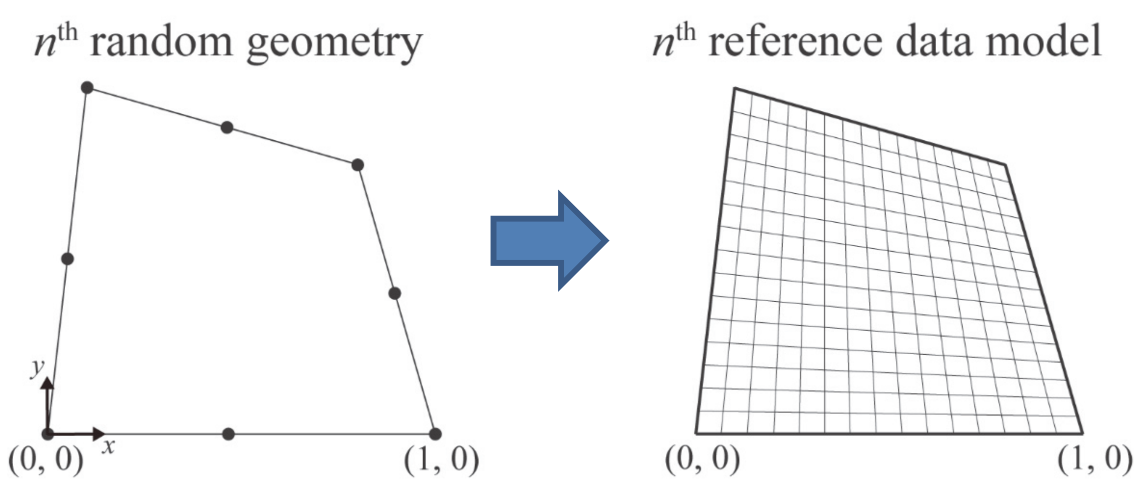
    1) interior angle < 10° or interior angle > 170°

    2) (maximum side length / minimum side length) > 10

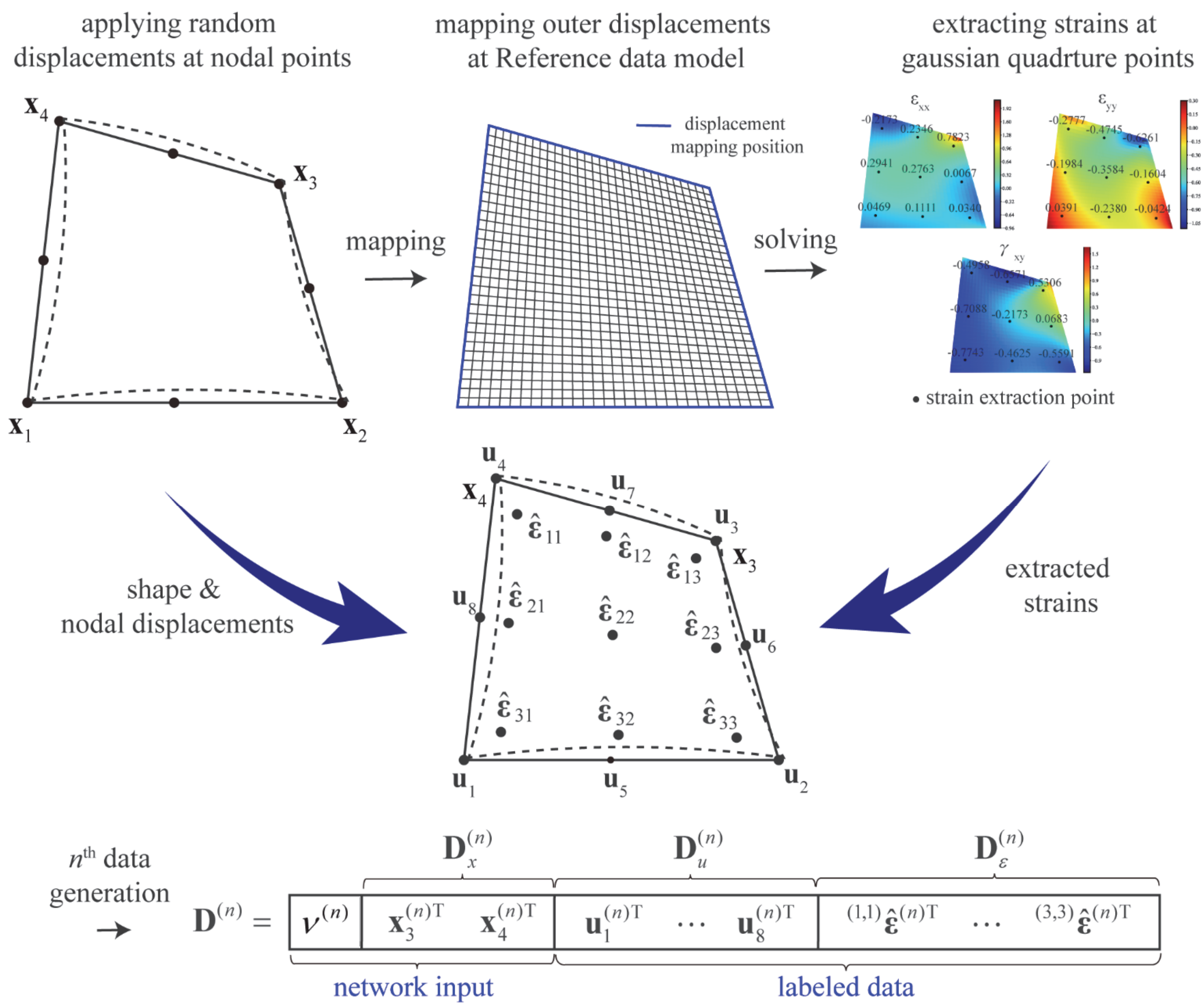


Data generation

- Generating strain data according to displacements, geometries, and Poisson's ratios
  - Random displacements and Poisson's ratios
    - The n^{th} nodal displacements ($\mathbf{u}_i^{(n)}$) : randomly generated with a uniform distribution in the range of -0.25 to 0.25
    - The n^{th} Poisson's ratios ($\nu^{(n)}$) : randomly applied with a uniform distribution in the range of 0–0.499999999. ($E = 2.0 \times 10^{11}$)
  - Generation of reference data model
    - Reference data model : FEM model has the same geometry as element n with a uniform $N \times N$ mesh (N = 50 was used in this study)



Data generation



Network configuration and training

❖ Cost function

$$C(\boldsymbol{\theta}) = \frac{1}{27M} \sum_{n=1}^{M} \sum_{i=1}^{3} \sum_{j=1}^{3} \sum_{k=1}^{3} {}^{(i,j)}w \left| \frac{\sum_{l=1}^{16} \left({}^{(i,j)}_{\text{output}}\mathrm{b}_{kl}^{(n)} \left(\boldsymbol{\theta}, \nu^{(n)}, \mathbf{x}_3^{(n)\mathrm{T}}, \mathbf{x}_4^{(n)\mathrm{T}} \right) u_l^{(n)} \right) - {}^{(i,j)}\hat{\varepsilon}_k^{(n)}}{{}^{(i,j)}\hat{\varepsilon}_k^{(n)}} \right|$$

$\boldsymbol{\theta}$: network weights

M : the number of training data

${}^{(i,j)}_{\text{output}}\mathrm{b}_{kl}^{(n)}(\boldsymbol{\theta})$: network output (the k^{th} row and l^{th} column of ${}^{(i,j)}_{\text{output}}\mathbf{B}^{(n)}(\boldsymbol{\theta})$)

${}^{(i,j)}\hat{\varepsilon}_k^{(n)}$: reference strain data

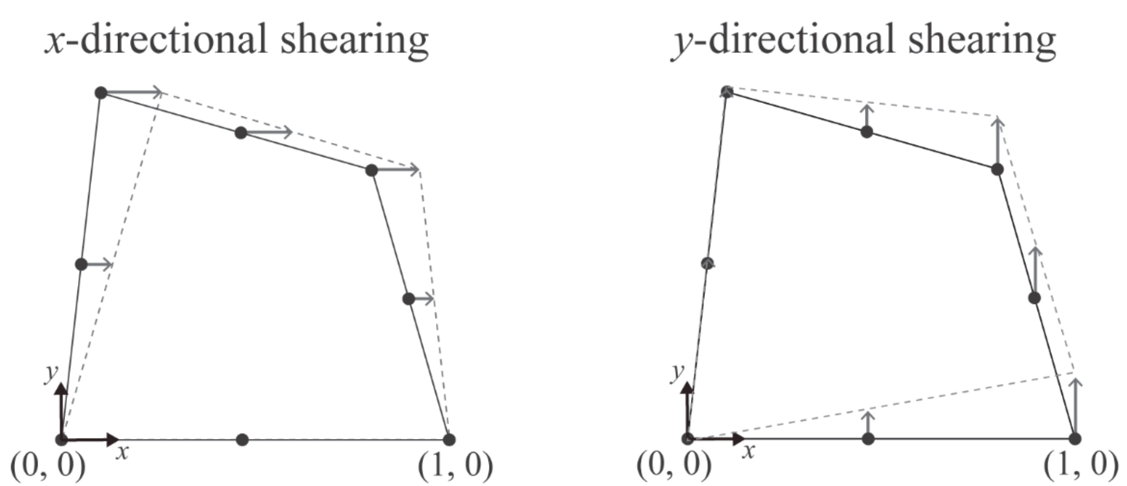
$u_l^{(n)}$: reference displacement data

❖ Data regulation

- Division by zero problem
  - When the strain value ${}^{(i,j)}\hat{\varepsilon}_k^{(n)}$ is close to 0, $C(\boldsymbol{\theta})$ increases sensitively.
  - Training data containing an absolute value less than 0.005 in the ${}^{(i,j)}\hat{\varepsilon}_k^{(n)}$ were excluded.
    → Almost pure shear, pure stretching data is also excluded.

Network configuration and training

- Data addition
  - Generating pure shear data on the normalized random geometry.
    - Displacements of x-directional shearing data : $\mathbf{u}_i^{(n)} = \begin{bmatrix} y_i^{(n)} & 0 \end{bmatrix}^{\mathrm{T}}$
    - Displacements of y-directional shearing data : $\mathbf{u}_i^{(n)} = \begin{bmatrix} 0 & x_i^{(n)} \end{bmatrix}^{\mathrm{T}}$
    - The zero-strain components → 0.5% of the maximum strain component in each data.



- Consideration of rigid body motion
  - finite elements should produce zero-strain energy under rigid body motion

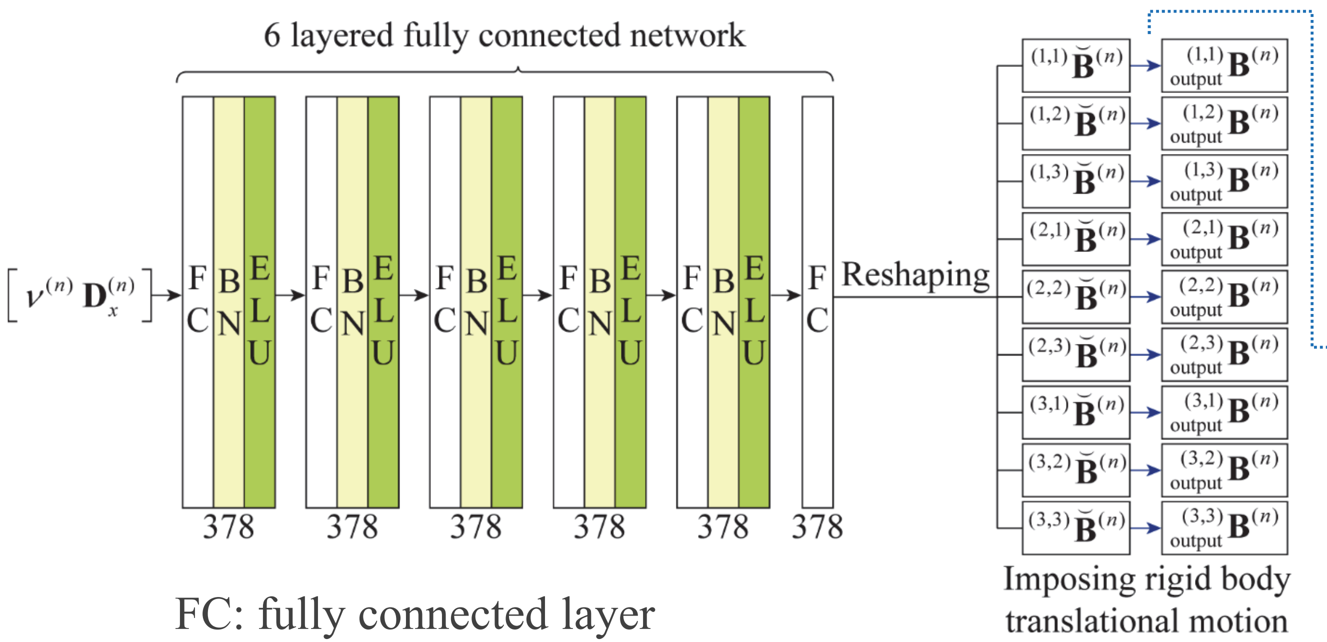
$${}^{(i,j)}_{\text{output}}\mathbf{B}^{(n)}(\boldsymbol{\theta})\Delta\mathbf{u} = \mathbf{0} \;\longrightarrow\; \sum_{l=1}^{8} {}^{(i,j)}_{\text{output}}\mathrm{b}_{kl}^{(n)}(\boldsymbol{\theta}) = 0 \;\text{ and }\; \sum_{l=9}^{16} {}^{(i,j)}_{\text{output}}\mathrm{b}_{kl}^{(n)}(\boldsymbol{\theta}) = 0 \;\text{ for }\; i,\, j,\, k=1,\,2,\,3$$

(rigid body translation)

Network configuration and training

❖ Network configuration

- Framework: Tensorflow



Matrix multiplication

$$
\begin{bmatrix}
1&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&-1\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1
\end{bmatrix}
$$

$$\sum_{l=1}^{8} {}^{(i,j)}_{\text{output}}\mathrm{b}_{kl}^{(n)}(\boldsymbol{\theta}) = 0 \qquad \sum_{l=9}^{16} {}^{(i,j)}_{\text{output}}\mathrm{b}_{kl}^{(n)}(\boldsymbol{\theta}) = 0$$

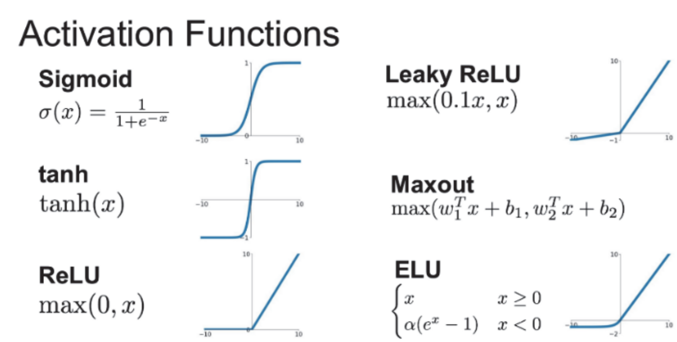
FC: fully connected layer

BN: batch normalization

ELU: exponential linear unit

${}^{(i,j)}\breve{\mathbf{B}}^{(n)}(\boldsymbol{\theta})$: pre B-matrix

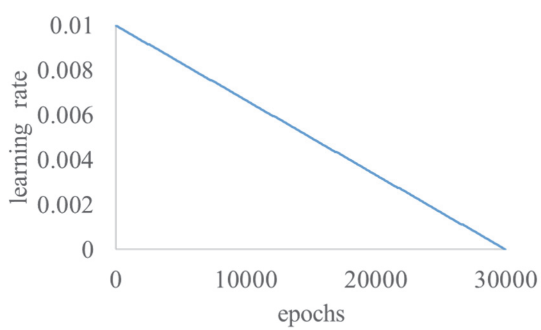
${}^{(i,j)}_{\text{output}}\mathbf{B}^{(n)}(\boldsymbol{\theta})$: generated B-matrix

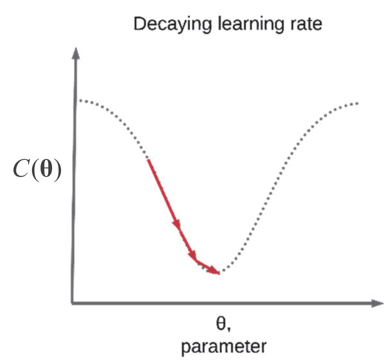


Network configuration and training

❖ Training configuration

- The number of training data : 300,000 (random : 294,000 + pure shear : 6000).
- The number of test data : 50,000 (random : 48,800 + pure shear : 1200).
- Optimizer : Adam optimizer.
- Weight initialization method : Xavier initializer.
- The number of epochs : 30,000.
- Batch size : 50,000.
- learning rate decaying: 0.01 → 0.



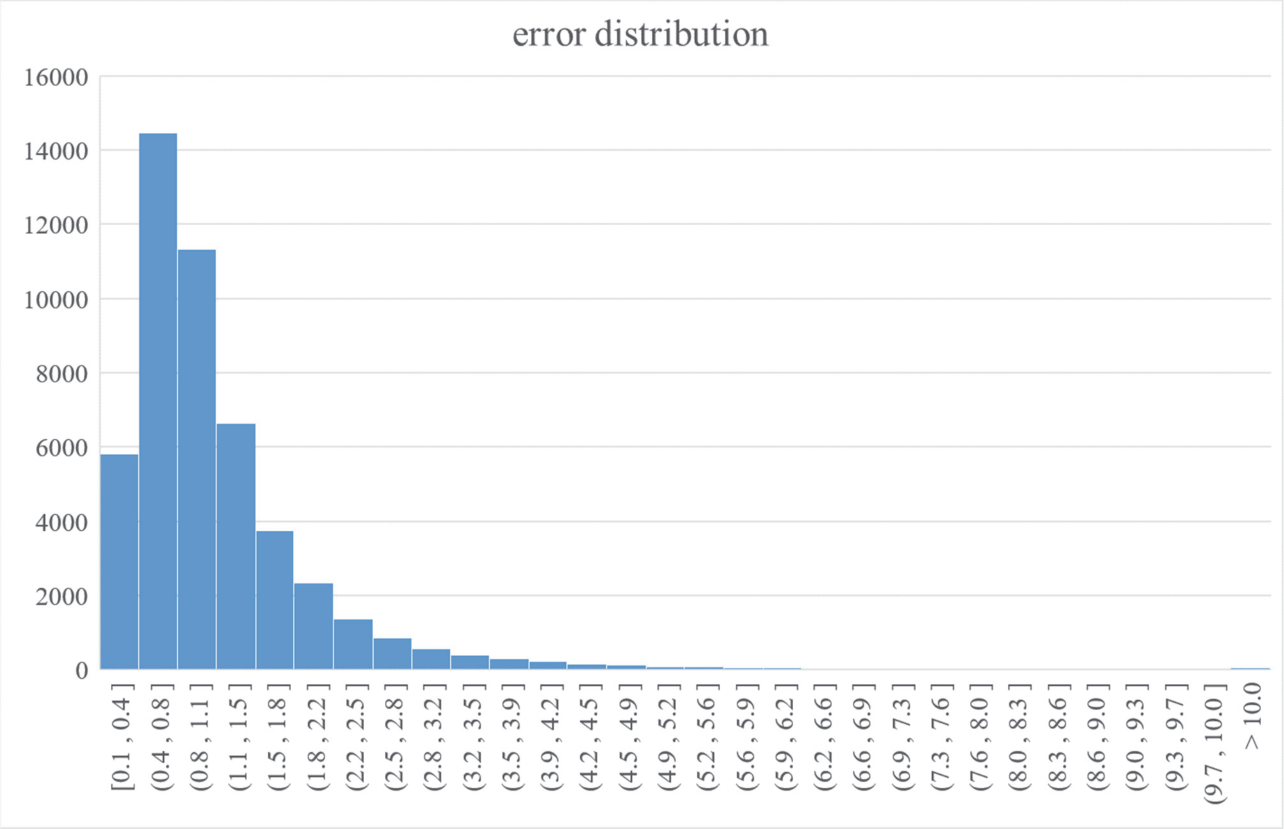


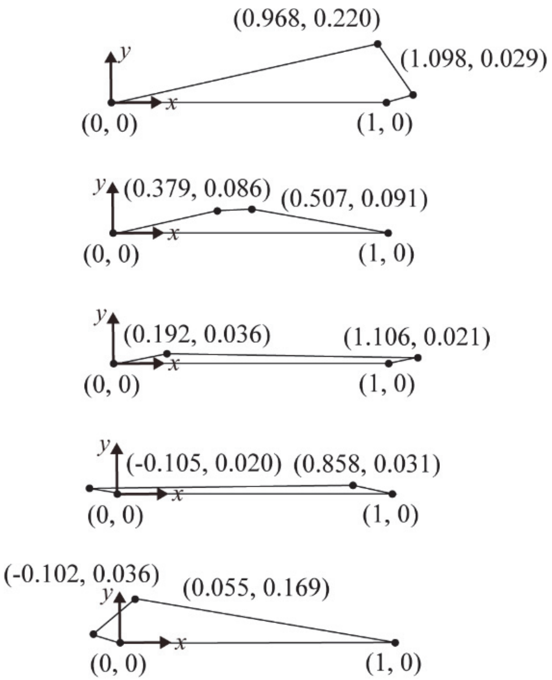
❖ Training result

training data error (%)	test data error (%)
1.24	1.67

Network configuration and training

- Error distribution in the pure random test data
  - The number of test data : 48,800
  - Average of the error: 1.11%
  - Standard deviation of the error : 1.12%



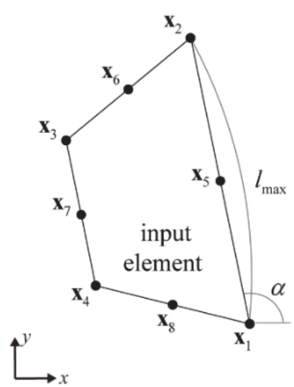


Construction of the stiffness matrix

❖ Pre-processing of the network input

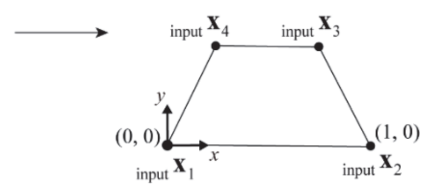
1) Elemental connectivity modification
   - The side length between $\mathbf{x}_1$ and $\mathbf{x}_2$ should be the longest.

2) Translation, rotation, and resizing of the nodal coordinates of the element



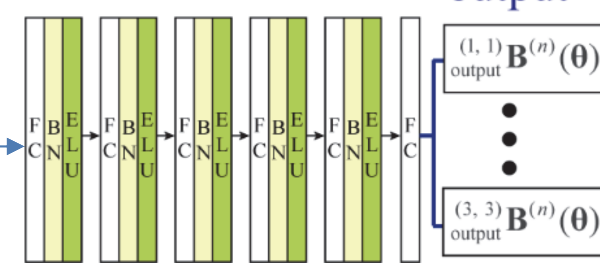
Translation, rotation & resizing

$$ {}_{\text{input}}\mathbf{x}_i = \frac{1}{l_{\max}} \mathbf{R}^{\mathrm{T}} (\mathbf{x}_i - \mathbf{x}_1) \quad \text{with } \mathbf{R} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} $$



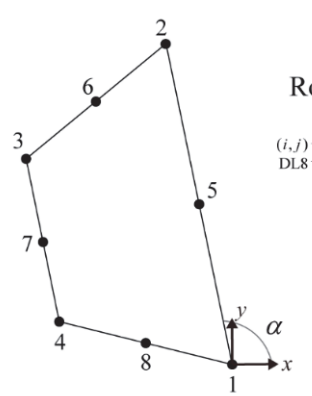
trained network

output



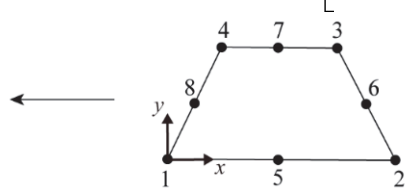
❖ Post-processing of the network output

1) Rotation, and resizing of the output



Rotation & resizing

$$ {}^{(i,j)}_{\text{DL8}}\mathbf{B} = \frac{1}{l_{\max}} \mathbf{T}\, {}^{(i,j)}_{\text{output}}\mathbf{B}\mathbf{Q}^{\mathrm{T}} \quad \text{with } \mathbf{T} = \begin{bmatrix} \cos^2\alpha & \sin^2\alpha & -\sin\alpha\cos\alpha \\ \sin^2\alpha & \cos^2\alpha & \sin\alpha\cos\alpha \\ 2\sin\alpha\cos\alpha & -2\sin\alpha\cos\alpha & \cos^2\alpha - \sin^2\alpha \end{bmatrix}, \quad q_{kl} = \delta_{kl}\cos\alpha - \delta_{k(l-8)}\sin\alpha + \delta_{(k-8)l}\sin\alpha $$



Construction of the stiffness matrix

❖ Post-processing of the network output

2) Correction for patch test

- ${}^{(i,j)}_{\mathrm{DL8}}\mathbf{B}$ is approximated by the network. → hard to pass the patch tests.
- Applying B-bar method.

$$ {}^{(i,j)}_{\mathrm{DL8}}\overline{\mathbf{B}} = {}^{(i,j)}_{\mathrm{DL8}}\mathbf{B} + {}_{\mathrm{DL8}}\mathbf{B}' \quad \text{with} \quad {}_{\mathrm{DL8}}\mathbf{B}' = \frac{t}{V}\sum_{i=1}^{3}\sum_{j=1}^{3} {}^{(i,j)}w\left({}^{(i,j)}_{\mathrm{Q8}}\mathbf{B} - {}^{(i,j)}_{\mathrm{DL8}}\mathbf{B}\right){}^{(i,j)}J $$

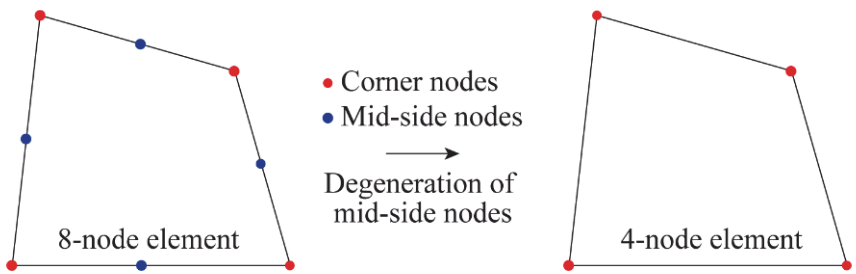
V : volume of element

❖ Stiffness matrix generation

$$ {}_{\mathrm{DL8}}\mathbf{K} = t\sum_{i=1}^{3}\sum_{j=1}^{3} {}^{(i,j)}w \, {}^{(i,j)}_{\mathrm{DL8}}\overline{\mathbf{B}}^{\mathrm{T}} \, \mathbf{C} \, {}^{(i,j)}_{\mathrm{DL8}}\overline{\mathbf{B}} \, {}^{(i,j)}J $$

4-node deep learned finite elements

- Applying incompatible element concept



  - Dividing the matrix to two parts

$$ {}^{(i,j)}_{\mathrm{DL8}}\mathbf{B}\mathbf{u} = \begin{bmatrix} {}^{(i,j)}\mathbf{B}_{\mathrm{C}} & {}^{(i,j)}\mathbf{B}_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathrm{C}} \\ \mathbf{u}_{\mathrm{I}} \end{bmatrix} $$

$\mathbf{u}_{\mathrm{C}}$: displacements of compatible nodes

$\mathbf{u}_{\mathrm{I}}$: displacements of incompatible nodes

  - Correction for patch test

$$ {}^{(i,j)}\overline{\mathbf{B}}_{\mathrm{C}} = {}^{(i,j)}\mathbf{B}_{\mathrm{C}} + \mathbf{B}'_{\mathrm{C}} \quad \text{and} \quad {}^{(i,j)}\overline{\mathbf{B}}_{\mathrm{I}} = {}^{(i,j)}\mathbf{B}_{\mathrm{I}} + \mathbf{B}'_{\mathrm{I}} $$

$$ \text{with} \quad \mathbf{B}'_{\mathrm{C}} = \frac{t}{V}\sum_{i=1}^{3}\sum_{j=1}^{3} {}^{(i,j)}w \left({}^{(i,j)}_{\mathrm{Q4}}\mathbf{B} - {}^{(i,j)}\mathbf{B}_{\mathrm{C}} \right) {}^{(i,j)}J \,, \quad \mathbf{B}'_{\mathrm{I}} = -\frac{t}{V}\sum_{i=1}^{3}\sum_{j=1}^{3} {}^{(i,j)}w \, {}^{(i,j)}\mathbf{B}_{\mathrm{I}} \, {}^{(i,j)}J $$

  - Stiffness matrix generation

$$ \mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathrm{CC}} & \mathbf{K}_{\mathrm{CI}} \\ \mathbf{K}_{\mathrm{IC}} & \mathbf{K}_{\mathrm{II}} \end{bmatrix} = t\sum_{i=1}^{3}\sum_{j=1}^{3} {}^{(i,j)}w \, {}^{(i,j)}_{\mathrm{DL4}}\mathbf{B}^{\mathrm{T}} \mathbf{C} \, {}^{(i,j)}_{\mathrm{DL4}}\mathbf{B} \, {}^{(i,j)}J \quad \text{with} \quad {}^{(i,j)}_{\mathrm{DL4}}\mathbf{B} = \begin{bmatrix} {}^{(i,j)}\overline{\mathbf{B}}_{\mathrm{C}} & {}^{(i,j)}\overline{\mathbf{B}}_{\mathrm{I}} \end{bmatrix} $$

$$ {}_{\mathrm{DL4}}\mathbf{K} = \mathbf{K}_{\mathrm{CC}} - \mathbf{K}_{\mathrm{CI}} \left(\mathbf{K}_{\mathrm{II}} \right)^{-1} \mathbf{K}_{\mathrm{IC}} $$

Zero energy mode tests

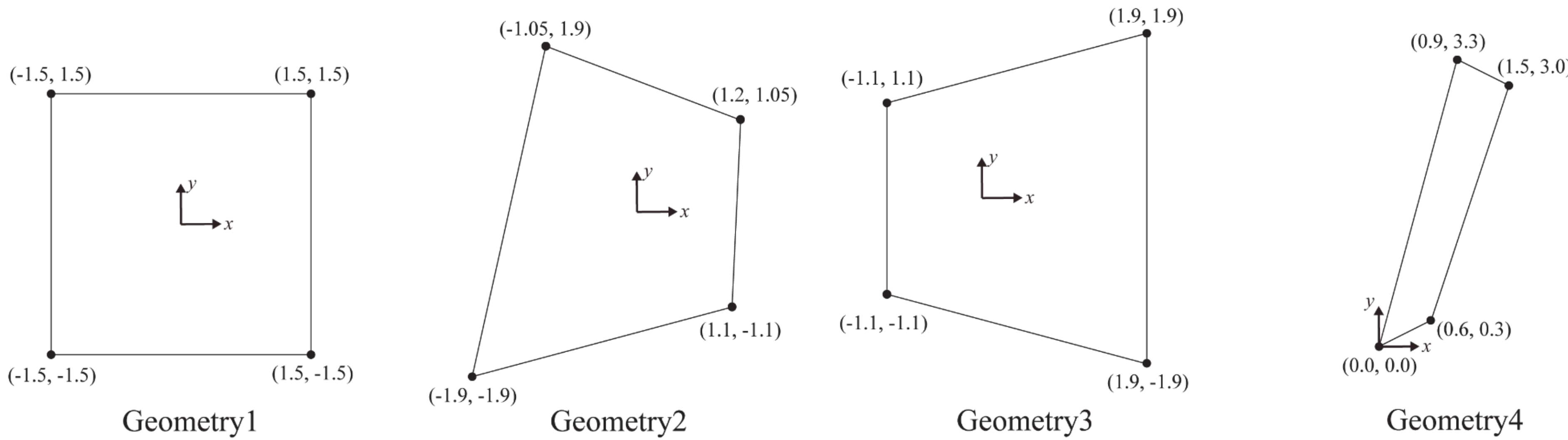
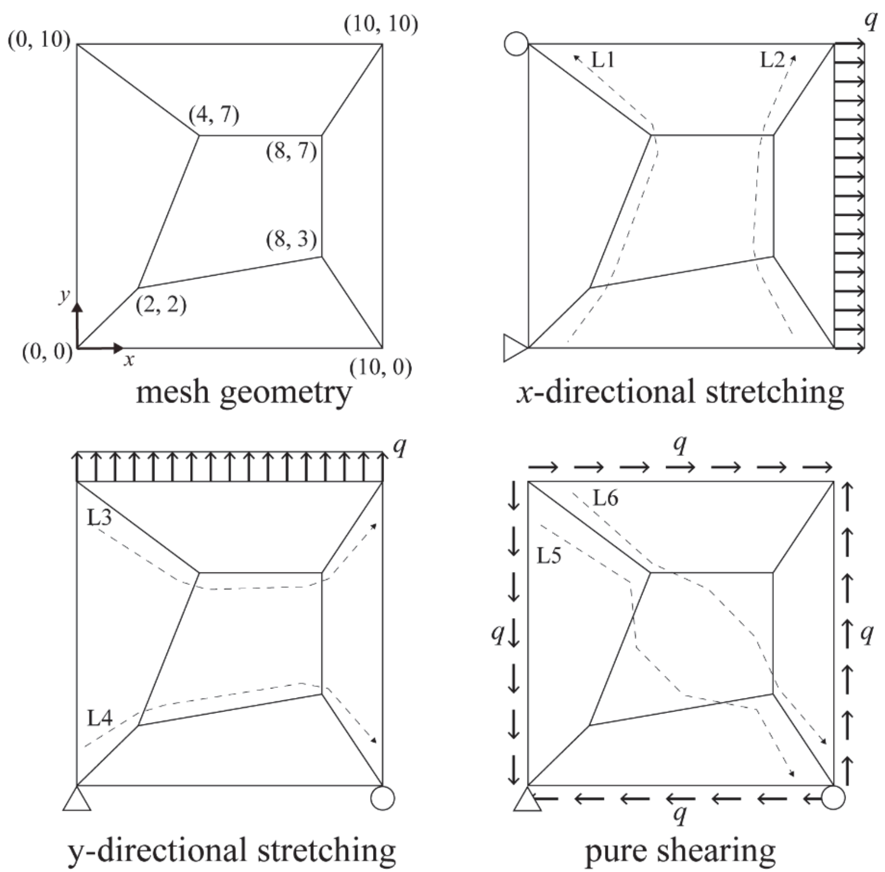


Table. **Eigenvalues corresponding to the 1st~6th modes** ($E = 1.5 \times 10^3$, $\upsilon = 0.3$, thickness = 1.0)

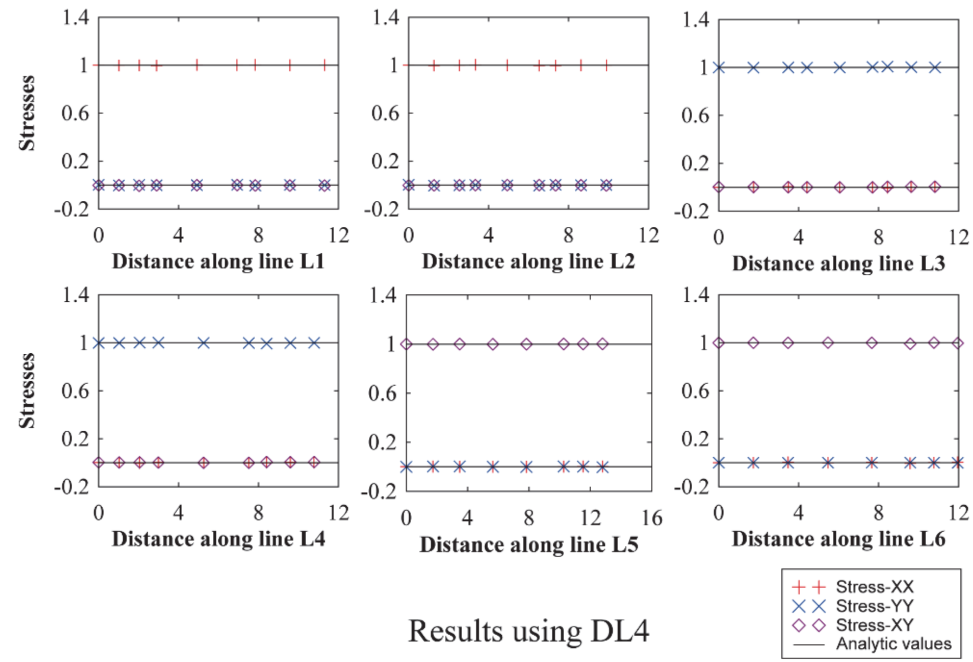
Mode	Geometry 1		Geometry 2		Geometry 3		Geometry 4	
	DL8	**DL4**	**DL8**	**DL4**	**DL8**	**DL4**	**DL8**	**DL4**
1	5.88E-14	4.82E-13	3.32E-13	1.83E-13	1.31E-13	8.31E-13	2.45E-13	7.93E-13
2	3.83E-13	9.43E-13	4.84E-13	7.84E-12	6.68E-13	1.25E-12	7.50E-13	2.86E-12
3	3.36E-03	1.31E-03	2.24E-04	4.87E-03	1.04E-04	5.01E-03	1.44E-03	2.33E-03
4	4.16E+02	1.15E+03	4.33E+02	4.36E+02	1.96E+02	4.69E+02	1.80E+01	7.41E+01
5	4.19E+02	1.15E+03	4.71E+02	5.25E+02	3.87E+02	5.02E+02	1.04E+02	2.73E+02
6	5.07E+02	4.97E+03	4.98E+02	1.14E+03	4.92E+02	1.18E+03	1.23E+02	2.12E+03

Patch tests

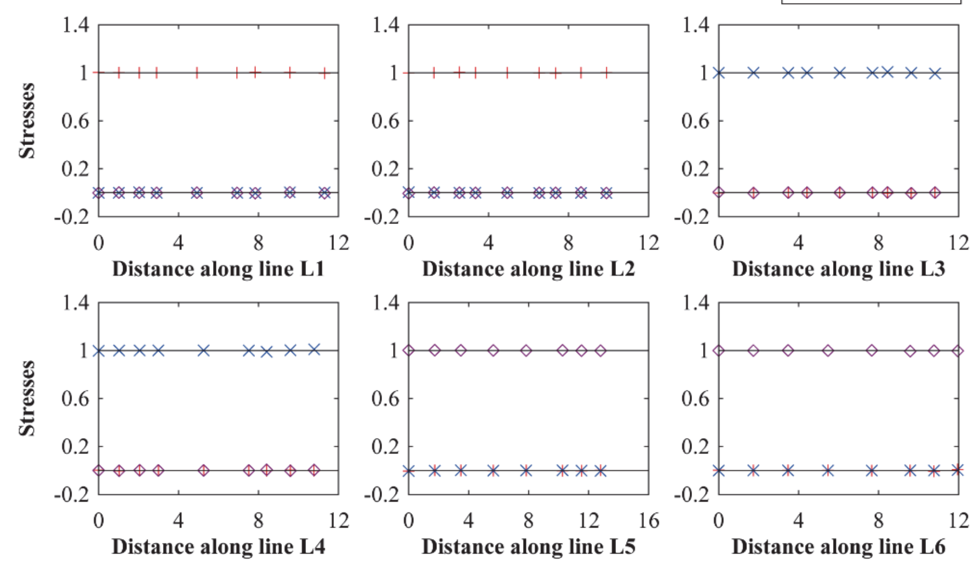


($q = 1.0$, $E = 3.0\times10^{7}$, $\nu = 0.3$, thickness = 1.0)

Results using DL8



Results using DL4



Numerical examples

❖ Elements to be compared

- Linear elements
  - Q4 : standard 4-node quadrilateral element
  - QM6 : 4-node quadrilateral element with incompatible modes
  - P182 : 4-node quadrilateral element in ANSYS 18.0
  - DL4 : Deep learned 4-node quadrilateral element
- Quadratic elements
  - Q8 : standard 8-node quadrilateral element
  - Q9 : standard 9-node quadrilateral element
  - P183 : 8-node quadrilateral element in ANSYS 18.0
  - DL8 : Deep learned 8-node quadrilateral element

❖ Strain energy error for convergence curve

$$E_e = \left| \frac{E_{\mathrm{ref}} - E_h}{E_{\mathrm{ref}}} \right|$$

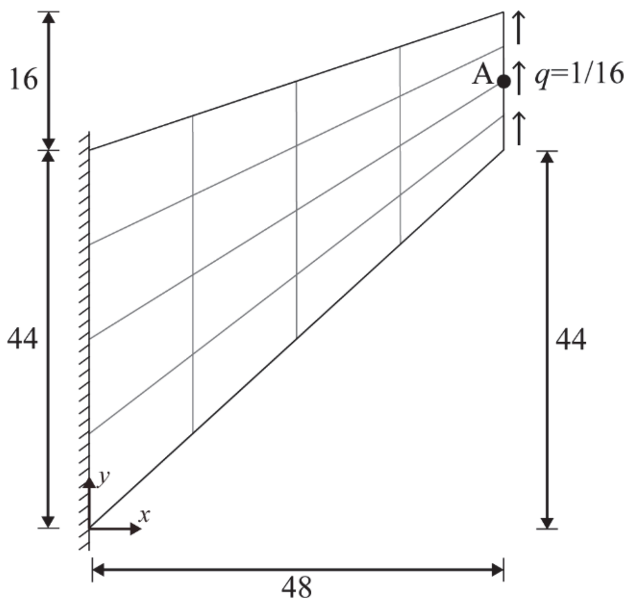
E_{ref} : strain energy obtained from the reference solution

E_h : strain energy obtained from the finite element solution

Numerical examples

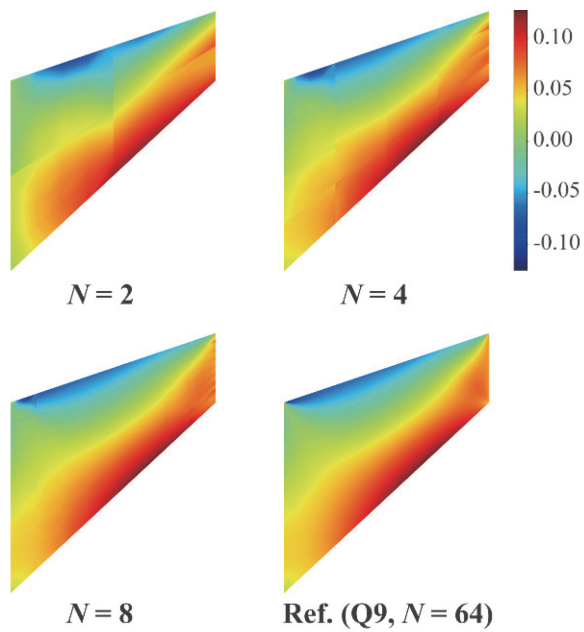
❖ Cook's skew beam problem

Problem description



($E = 3.0\times10^7$, $\nu = 0.3$, thickness = 1.0)

DL8 Stress distribution



Convergence curves

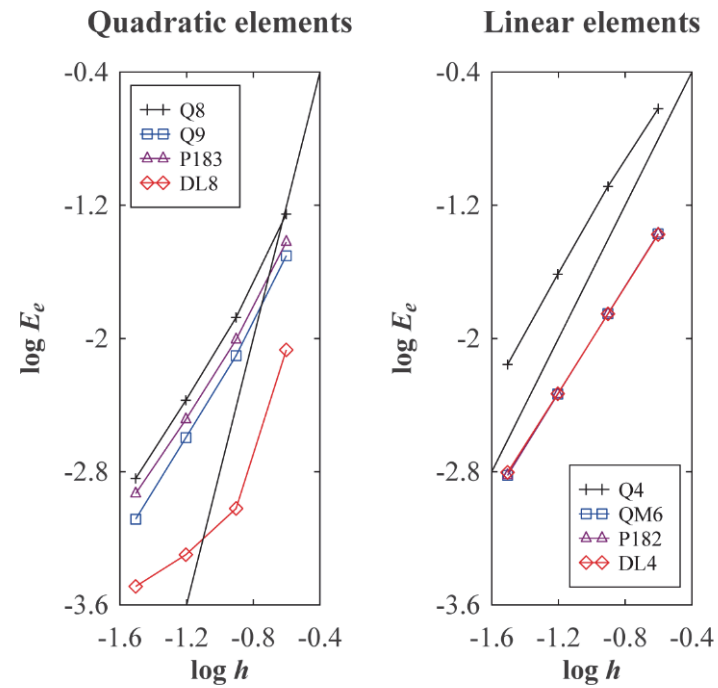


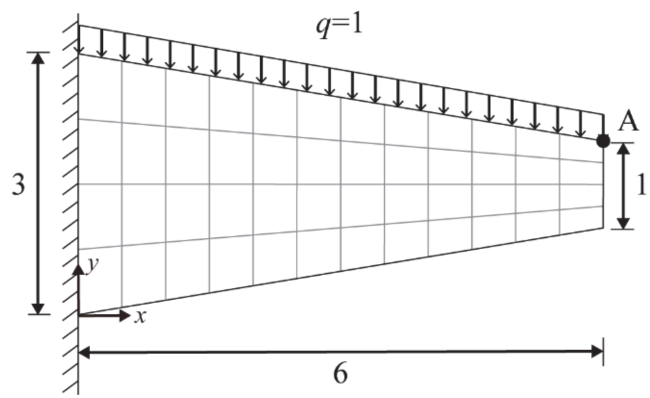
Table. Normalized deflections at point A (reference solution: 23.9662).

Mesh	Quadratic elements				Linear elements			
	Q8	Q9	P183	DL8	Q4	QM6	P182	DL4
2×2	0.9479	0.9717	0.9668	**0.9868**	0.4942	**0.8783**	**0.8783**	0.8756
4×4	0.9892	0.9947	0.9900	**0.9992**	0.7635	0.9604	0.9604	**0.9606**
8×8	0.9966	0.9983	0.9967	**0.9995**	0.9213	**0.9884**	**0.9884**	0.9883
16×16	0.9987	0.9993	0.9989	**0.9996**	0.9776	**0.9965**	**0.9965**	0.9964

Numerical examples

❖ Tapered beam problem

Problem description



($E = 3.0\times10^2$, $\nu = 0.3$, thickness = 0.1)

Convergence curves

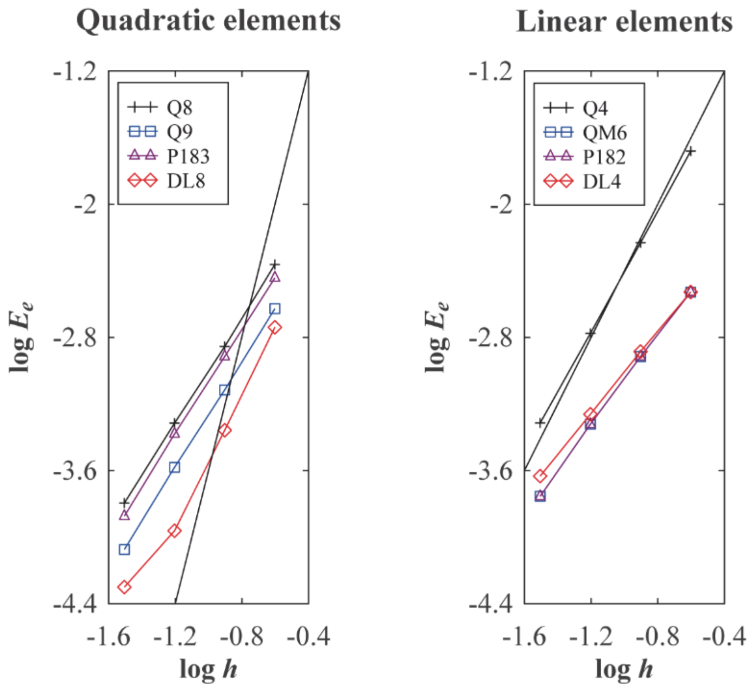


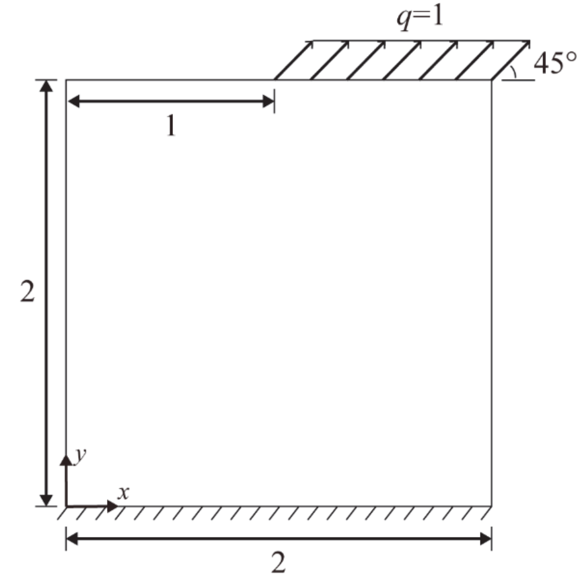
Table. Normalized deflections at point A (reference solution: 0.49679).

Mesh	Quadratic elements				Linear elements			
	Q8	Q9	P183	DL8	Q4	QM6	P182	DL4
2×6	0.9976	0.9987	0.9982	**0.9990**	0.9214	**1.0006**	**1.0006**	1.0027
4×12	0.9992	0.9995	0.9994	**0.9998**	0.9781	0.9998	0.9998	**0.9999**
8×24	0.9997	0.9998	0.9998	**0.9999**	0.9941	**0.9997**	**0.9997**	**0.9997**
16×48	0.9999	0.9999	0.9999	**1.0000**	0.9984	**0.9998**	**0.9998**	**0.9998**

Numerical examples

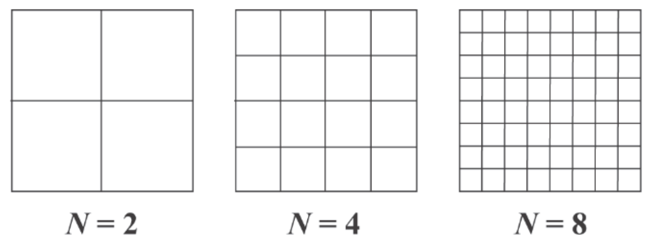
❖ Block problem

Problem description

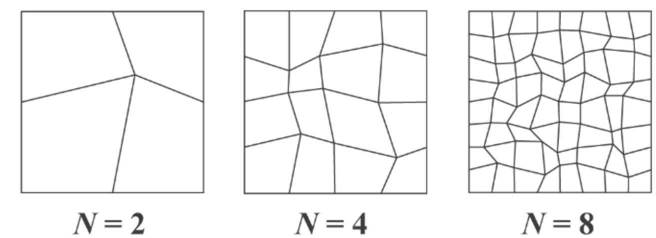


($E = 3.0\times10^{7}$, $\nu = 0.3$, thickness = 1.0)

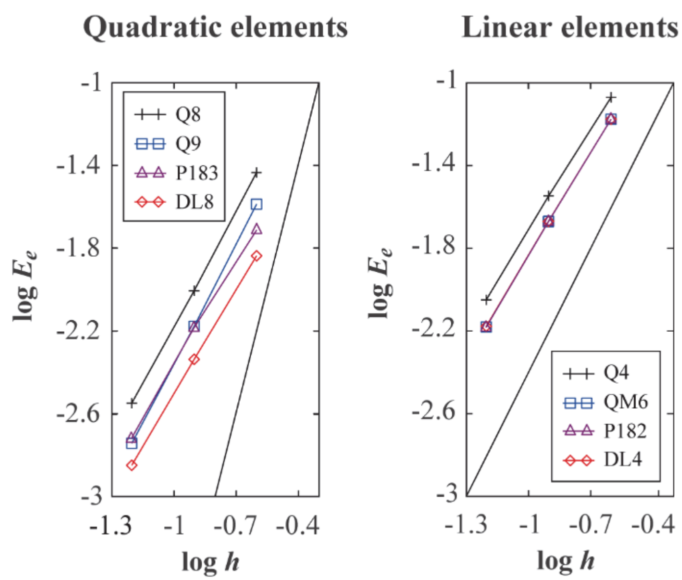
Regular mesh



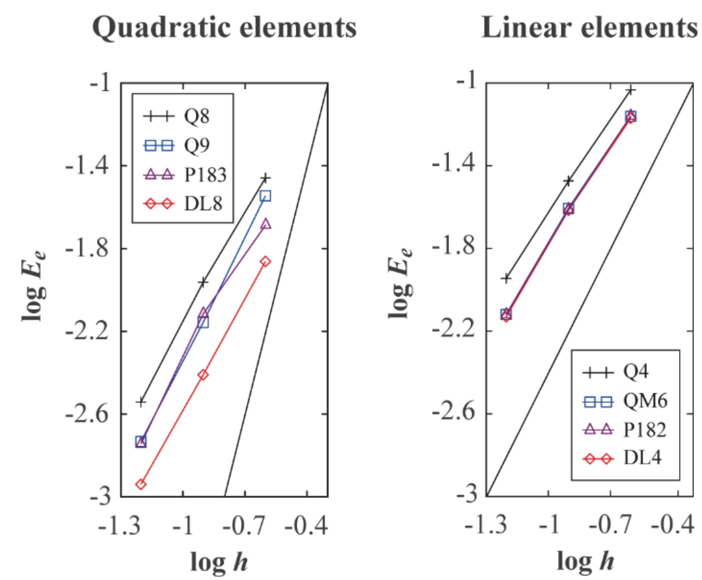
Distorted mesh



Convergence curves



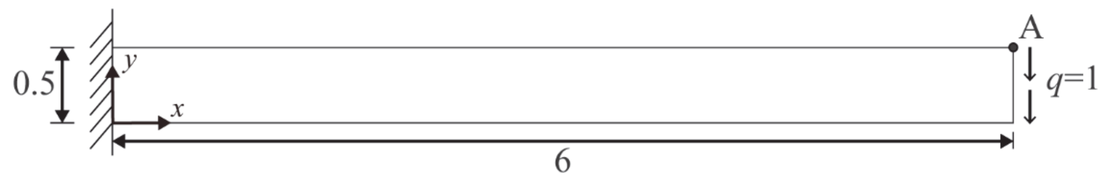
Convergence curves

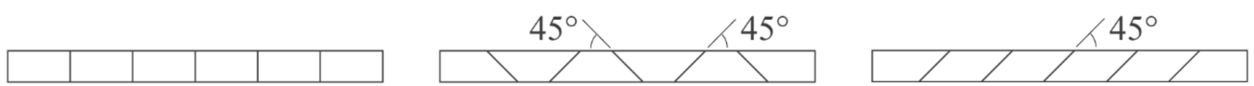


Numerical examples

❖ Cantilever beam problem

Problem description





Rectangular mesh   Trapzoidal mesh   Parallelogram mesh

($E = 1.0\times10^7$, $\nu = 0.3$, thickness = 0.1)

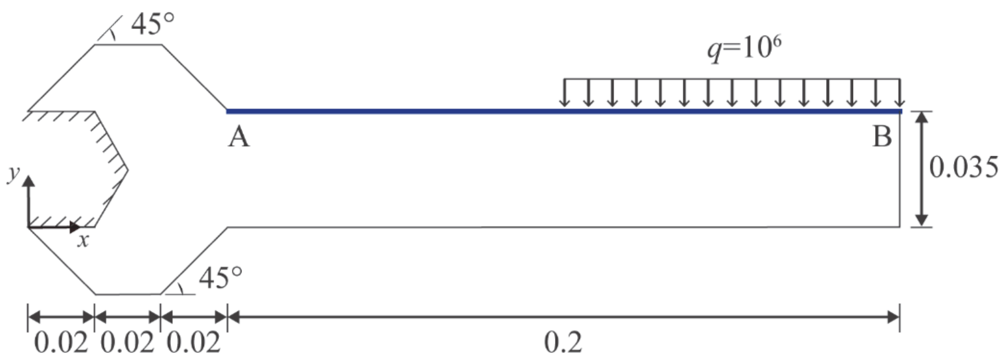
Table. Normalized deflections at point A (reference solution: -3.4694×10^{-3}).

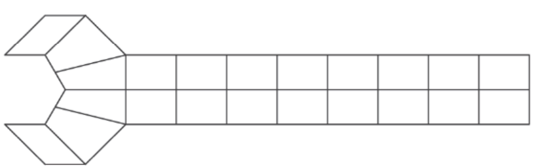
Mesh	Quadratic elements				Linear elements			
	Q8	**Q9**	**P183**	**DL8**	**Q4**	**QM6**	**P182**	**DL4**
Rectangular	0.9861	**0.9935**	0.9877	0.9862	0.3796	**0.9937**	**0.9937**	1.0115
Trapezoidal	0.8984	0.9877	0.9704	**0.9940**	0.1351	0.2064	0.2064	**0.2080**
Parallelogram	0.9888	0.9898	**0.9997**	0.9866	0.1492	**0.7932**	**0.7932**	0.7888

Numerical examples

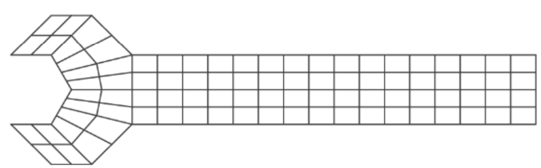
❖ Wrench problem

Problem description

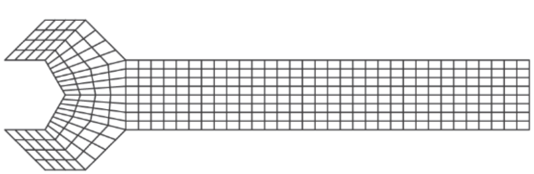




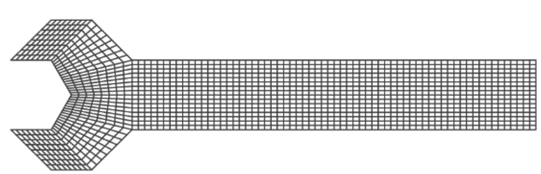
Coarse mesh



medium mesh



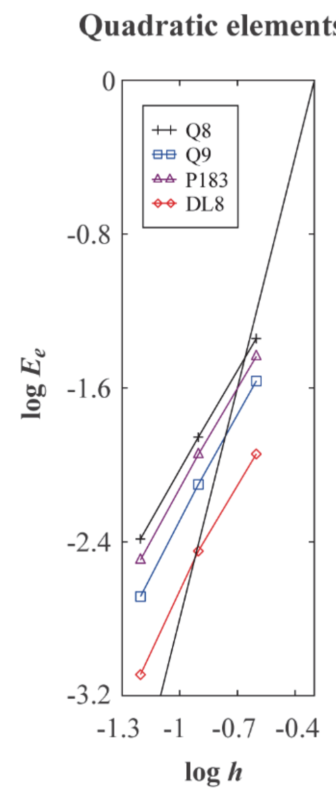
fine mesh

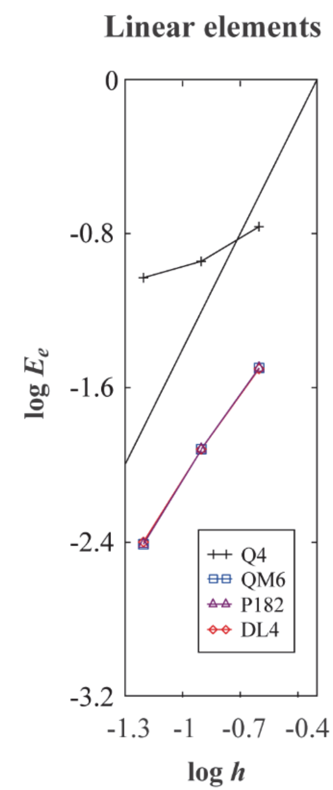


reference solution mesh

($E = 2.0\times10^{11}$, $\nu = 0.3$, thickness = 0.01)

Convergence curves

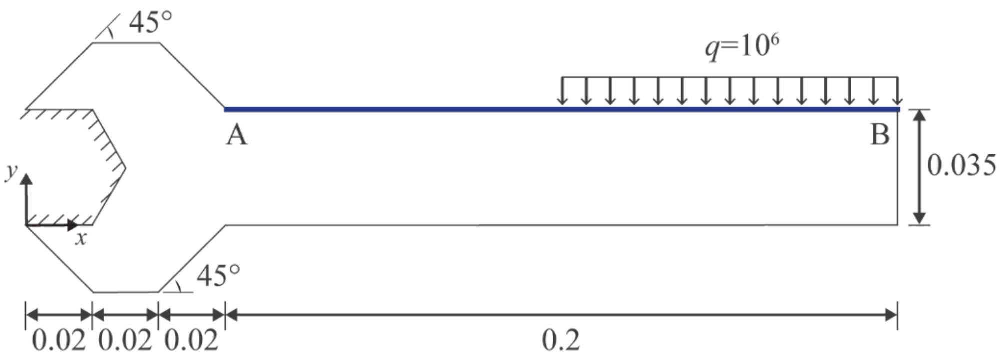


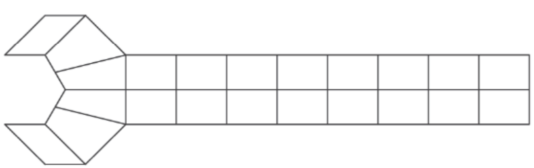


Numerical examples

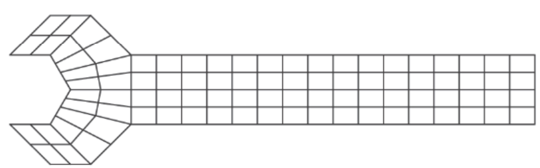
❖ Wrench problem

Problem description

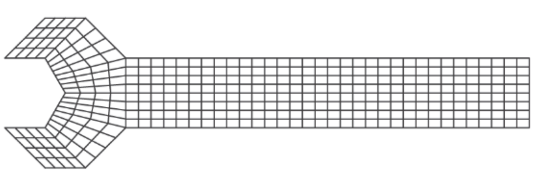




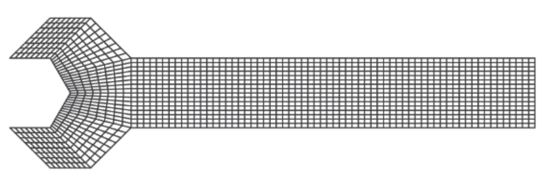
Coarse mesh



medium mesh



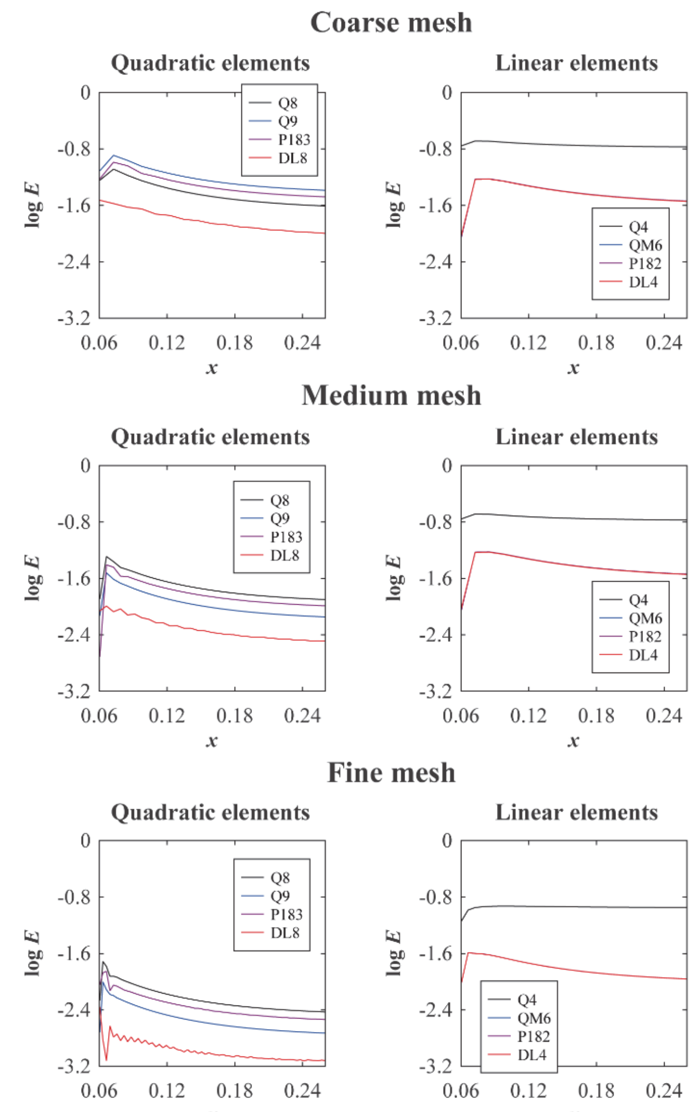
fine mesh



reference solution mesh

($E = 2.0\times10^{11}$, $\nu = 0.3$, thickness = 0.01)

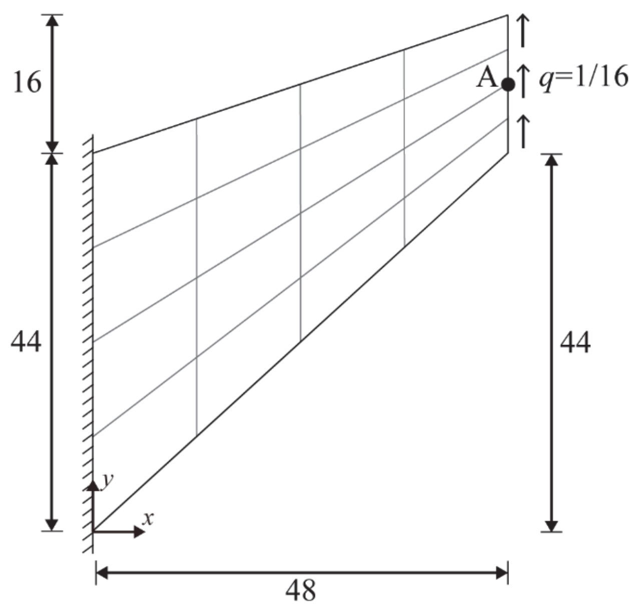
Vertical displacement error along the line AB



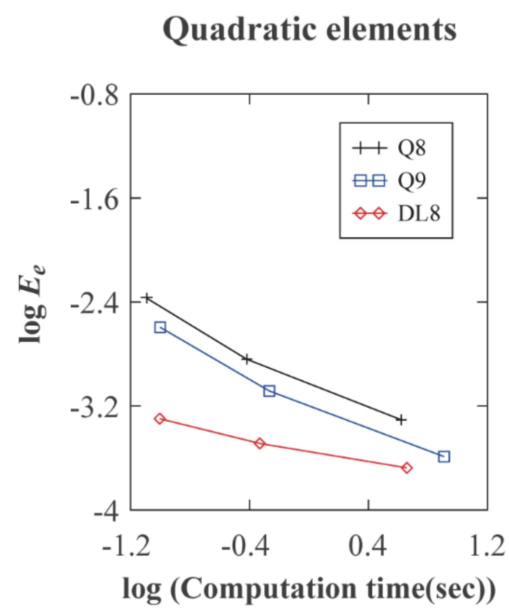
Computational efficiency

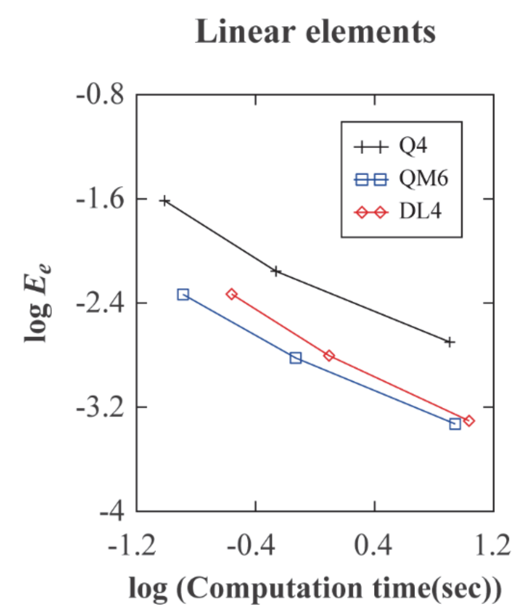
❖ Computational efficiency curves in the Cook's skew beam problem

Problem description



($E = 3.0\times10^7$, $\nu = 0.3$, thickness = 1.0)





Computation time : stiffness matrix generation + solving time (Intel(R) Core (TM) i7-2600 CPU @ 3.80 GHz, 12 GB memory, Microsoft Windows 10 64bit, Python environment)

- the DL8 element outperforms in the aspect of computational efficiency among the tested elements.

Closure (Topic 1)

- Deep learned finite elements are proposed
  - various new concepts and processes are presented
    - normalized element geometry
    - reference data model
    - pre-processing for the input
    - post-processing for the output
  - A way to make the developed elements better represent rigid body motions and constant strain fields are presented also.

- The performance of the developed elements was evaluated through various numerical examples
  - DL8 showed promising ability in both accuracy and computational efficiency.

- Limitations of DLFE
  - Geometrical limit (out of train data distribution, see Appendix C)
  - Problem with increasing DOF

Topic 2. Self-Updated Finite Element

Overview of SUFE

❖ Target element

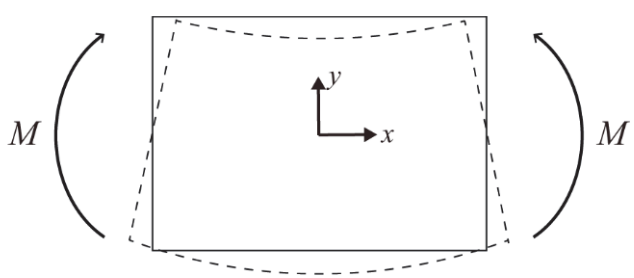
- 2D 4-node plane stress elements

❖ Objectives

- Generating adaptive B-matrix without any mesh change using displacements to remove shear locking.

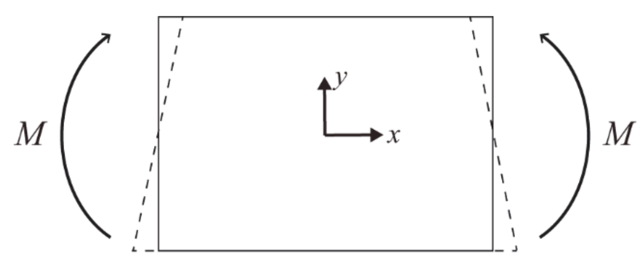
❖ Shear locking

- In certain cases, the displacements calculated by the finite element method are much smaller than they should be, and it is called locking.
- Shear locking occurs when elements are subjected to bending.



Exact displacements

$\gamma_{xy} = 0$

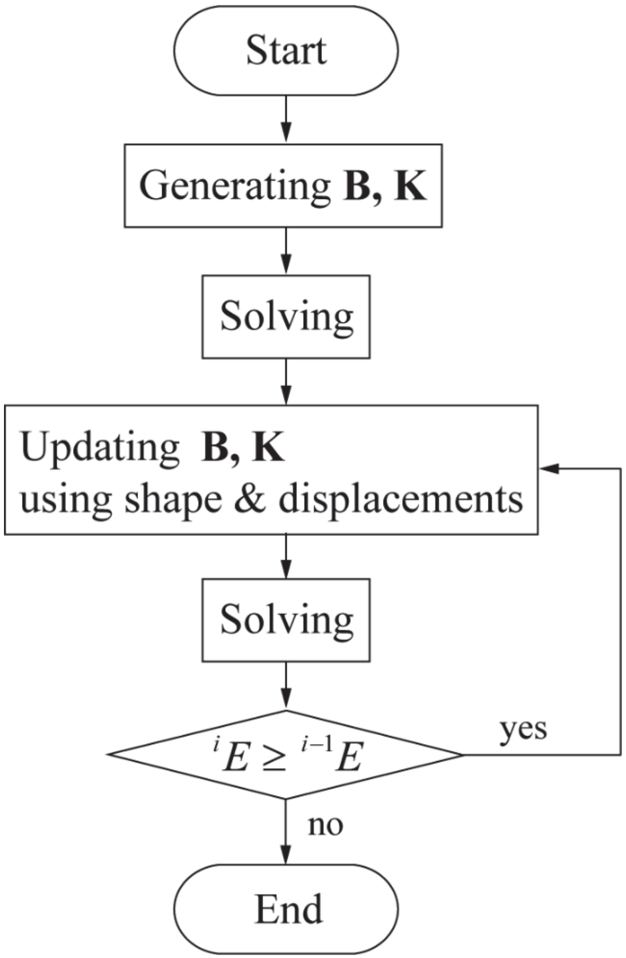


Linear element displacements

$\gamma_{xy} \neq 0$

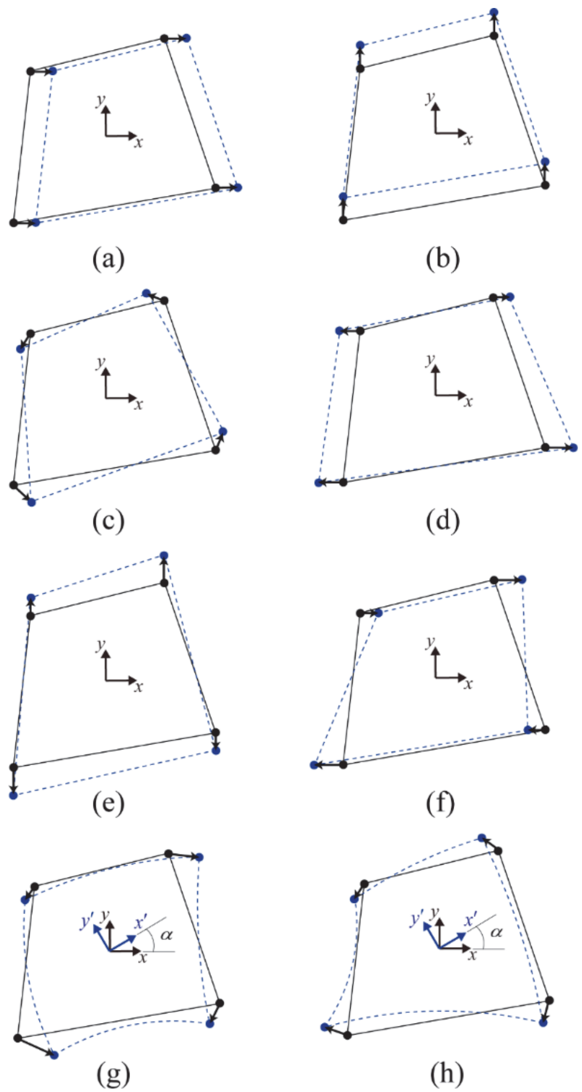
Overview of SUFE

❖ Concept



Mode based formulation

❖ Original element generation method used in SUFE



Kinematic modes of the linear element

$$\begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ v_4 \end{Bmatrix} = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots\cdots & \phi_{1,8} \\ \phi_{2,1} & \phi_{2,2} & \cdots\cdots & \phi_{2,8} \\ \vdots & \vdots & \cdots\cdots & \vdots \\ \phi_{8,1} & \phi_{8,2} & \cdots\cdots & \phi_{8,8} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_8 \end{Bmatrix} \quad \text{or} \quad \mathbf{U} = \boldsymbol{\psi}\,\mathbf{Q}$$

$$\boldsymbol{\varepsilon} = \mathbf{BU} = \mathbf{B}\boldsymbol{\psi}\mathbf{Q} = \hat{\mathbf{B}}\mathbf{Q}$$

$$\hat{\mathbf{B}} = \left[\begin{array}{cccccc|cc} 0 & 0 & 0 & 1 & 0 & 0 & e_1^7 & e_1^8 \\ 0 & 0 & 0 & 0 & 1 & 0 & e_2^7 & e_2^8 \\ 0 & 0 & 0 & 0 & 0 & 1 & e_3^7 & e_3^8 \end{array}\right]$$

assumed strains according to local coordinates

$$\boldsymbol{\varepsilon} = \hat{\mathbf{B}}\,\mathbf{A} = \hat{\mathbf{B}}\,\boldsymbol{\psi}^{-1}\,\mathbf{U} = \mathbf{B}_m\mathbf{U}$$

Local coordinate determination rule

$\alpha = \underset{\alpha}{\text{minimize}}\, E$ where **U** is given

→ Optimization problem (too much time is needed)

Deep learning approach is needed

Data generation

❖ Generating α according to displacements, geometries, and Poisson's ratios

- Random geometry
  - Normalized geometry is used.
  - Severely distorted geometries are excluded
    1) interior angle < 1° or interior angle > 179°
    2) (maximum side length / minimum side length) > 100
- Random displacements and Poisson's ratio
  - Normalized by maximum displacement is used.
  - 5 independent displacements is extracted using 3 rigid body modes.
- Strain energy optimization to obtain α
  - Optimization algorithm: Nelder-Mead method
  - Initial seed: 30 seeds evenly distributed in the range of 0° to 90°
- Data set

$$\mathbf{D}^{(n)} = \Big[\underbrace{\nu^{(n)} \quad \mathbf{D}_x^{(n)} \quad \mathbf{D}_u^{(n)}}_{\text{Network input}} \quad \underbrace{\alpha^{(n)}}_{\text{Label}}\Big] \quad \text{with} \quad \mathbf{D}_x^{(n)} = \begin{bmatrix} \mathbf{x}_3^{(n)\mathrm{T}} & \mathbf{x}_4^{(n)\mathrm{T}} \end{bmatrix},$$

$$\mathbf{D}_u^{(n)} = \begin{bmatrix} \bar{u}_2^{(\mathrm{n})} & \bar{u}_3^{(\mathrm{n})} & \bar{u}_4^{(\mathrm{n})} & \bar{v}_3^{(n)} & \bar{v}_4^{(n)} \end{bmatrix}$$

Network configuration and training

❖ Cost function

$$C(\boldsymbol{\theta}) = \frac{1}{M}\sum_{n=1}^{M}\left|\alpha_o^{(n)}(\boldsymbol{\theta}) - \alpha^{(n)}\right|^{\frac{1}{2}}$$

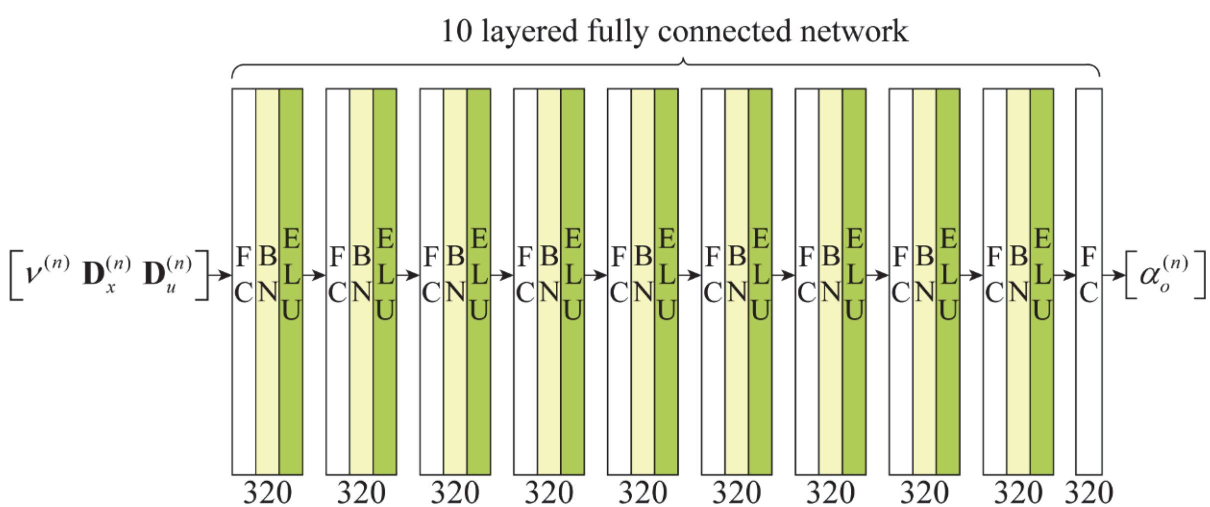
$\boldsymbol{\theta}$: network weights

M : the number of training data

$\alpha_o^{(n)}(\boldsymbol{\theta})$: network output

$\alpha^{(n)}$: reference α

❖ Network configuration

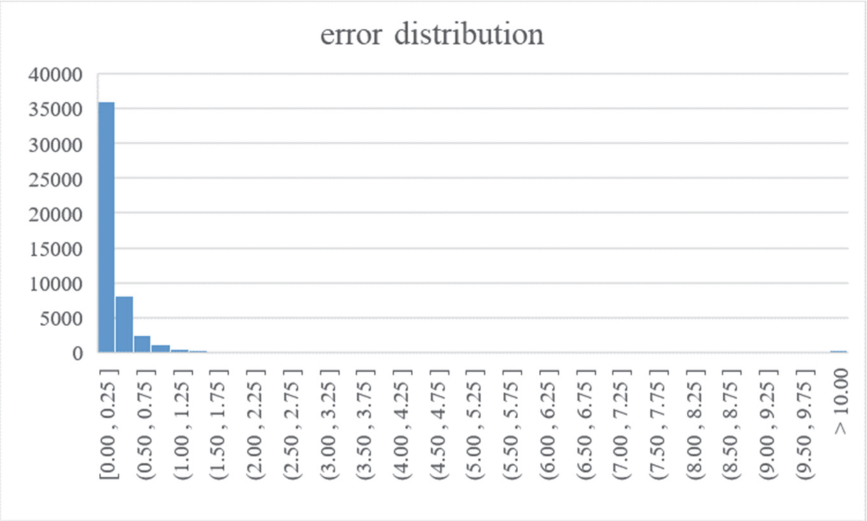


Network configuration and training

- Training configuration
  - The number of training data : 3,000,000
  - The number of test data : 50,000
  - Optimizer : Adam optimizer.
  - Weight initialization method : Xavier initializer.
  - The number of epochs : 30,000.
  - Batch size : 50,000.
  - learning rate decaying: 0.01 → 0.
- Training result

Cost function value of training data	Cost function value of test data
0.40	0.45

  - Average of the error: 0.44°
  - Standard deviation of the error : 2.49°



The stiffness matrix generation

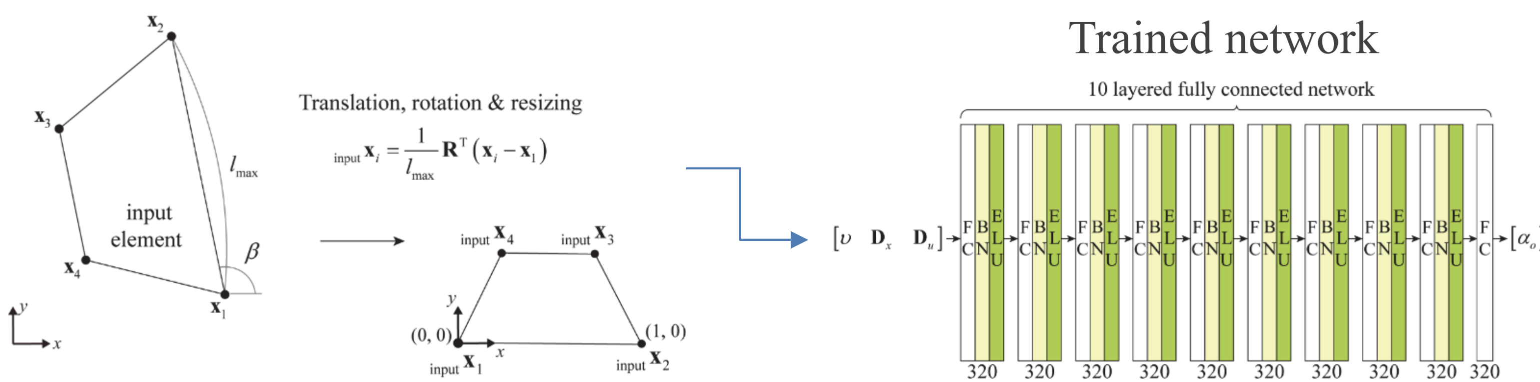
❖ Pre-processing of the network input

1) Elemental connectivity modification

   - The side length between $\mathbf{x}_1$ and $\mathbf{x}_2$ should be the longest.

2) Nodal displacements rotation and normalize $\rightarrow$ $\mathbf{D}_u = \begin{bmatrix} \bar{u}_2 & \bar{u}_3 & \bar{u}_4 & \bar{v}_3 & \bar{v}_4 \end{bmatrix}$

3) Translation, rotation, and resizing of the nodal coordinates of the element



❖ Post-processing of the network output

- Output modification

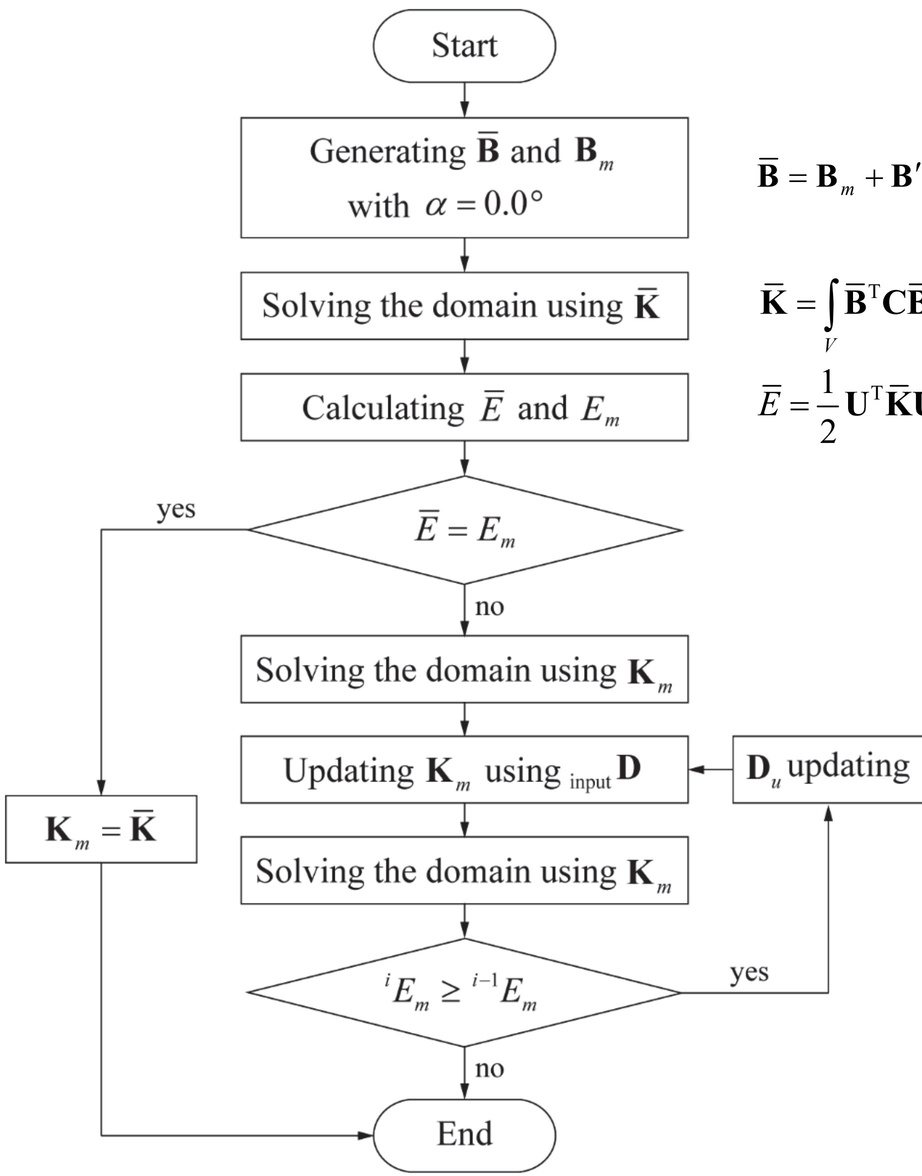
$$\alpha = \alpha_o + \beta$$

❖ the stiffness matrix generation

- $\mathbf{B}_m$, $\mathbf{K}_m$ generation

The stiffness matrix updating procedures

❖ Iteration procedures

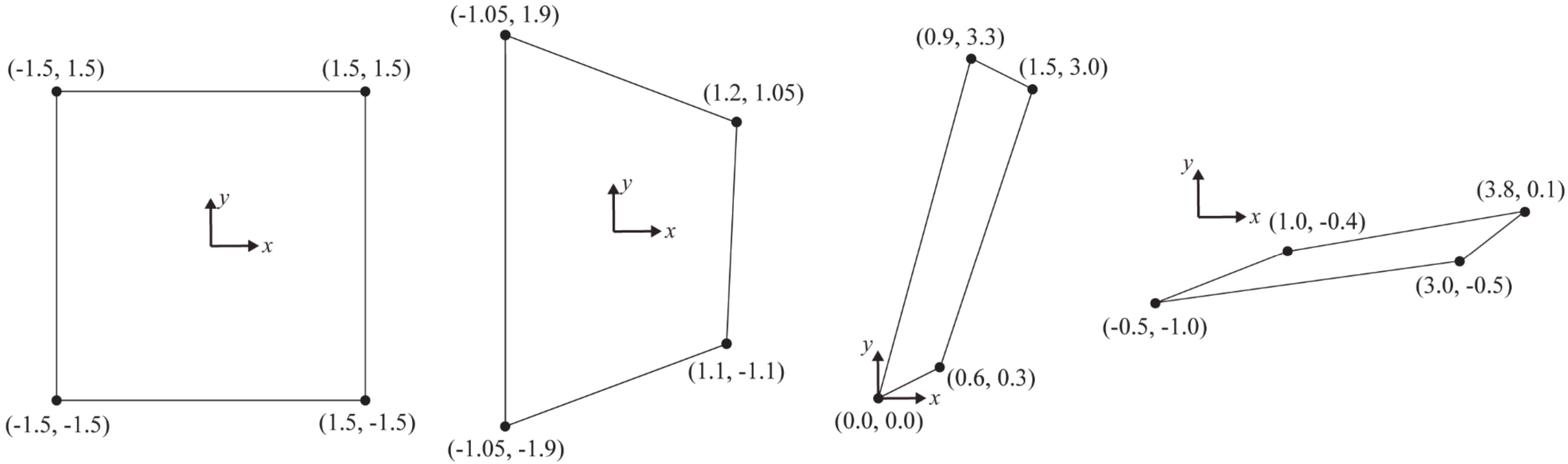


$$\overline{\mathbf{B}} = \mathbf{B}_m + \mathbf{B}' \quad \text{with} \quad \mathbf{B}' = -\frac{1}{V}\int_V (\mathbf{B}_m - \mathbf{B}_{Q4})\,dV$$

$$\overline{\mathbf{K}} = \int_V \overline{\mathbf{B}}^{\mathrm{T}} \mathbf{C} \overline{\mathbf{B}}\,dV$$

$$\overline{E} = \frac{1}{2}\mathbf{U}^{\mathrm{T}}\overline{\mathbf{K}}\mathbf{U}\,, \quad E_m = \frac{1}{2}\mathbf{U}^{\mathrm{T}}\mathbf{K}_m\mathbf{U} \quad \text{with} \quad \mathbf{K}_m = \int_V \mathbf{B}_m^{\mathrm{T}}\mathbf{C}\mathbf{B}_m\,dV$$

Zero energy mode tests



Geometry1 Geometry2 Geometry3 Geometry4

Table. Eigenvalues corresponding to the 1st~6th modes ($E = 1.5\times10^3$, $\nu = 0.3$, thickness = 1.0 **)**

Mode	Geometry 1	Geometry 2	Geometry 3	Geometry 4
1	-2.76E-14	5.58E-14	2.37E-14	1.06E-12
2	-6.12E-14	1.48E-13	-4.25E-14	-3.66E-12
3	-8.60E-14	-1.00E-13	7.37E-13	9.96E-12
4	5.00E+02	4.04E+02	2.61E+02	1.94E+01
5	5.00E+02	8.43E+02	3.18E+02	1.66E+02
6	1.15E+03	1.06E+03	2.62E+03	5.06E+03

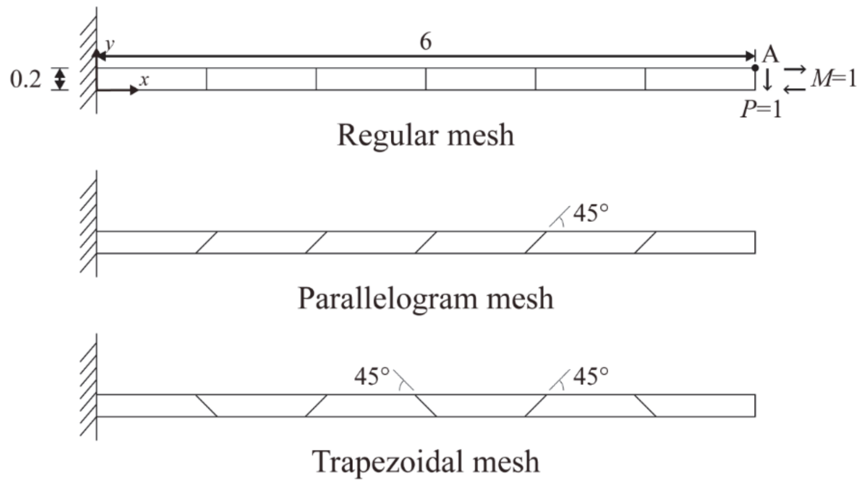
Numerical examples

❖ Elements to be compared

Symbol	Description
Q4	Standard 4-node quadrilateral element
QM6	4-node quadrilateral element with incompatible modes
P-S	4-node hybrid stress element
SPS	4-node hybrid stress elements with adjustable parameters
$\bar{B}$-QE4	4-node assumed strain element (B-bar)
QACM4	4-node incompatible element using QACM-I
NQ6	4-node incompatible hybrid stress element
NQ10	4-node incompatible hybrid stress element
GC-Q6	4-node generalized conforming element
PEAS7	4-node assumed strain element
2D-MITC4/1	4-node assumed strain element based on the MITC method

Numerical examples

❖ MacNeal's thin cantilever beam



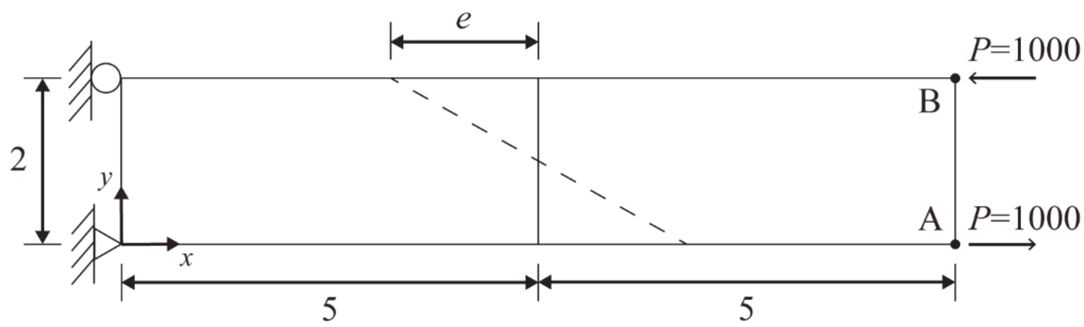
($E = 1.0 \times 10^7$, $\nu = 0.3$, thickness = 0.1)

Table. Normalized vertical displacements at point A in the MacNeal′s thin cantilever beam using different meshes, data in bold are the results obtained by the elements proposed in this study, and the number in bracket is the number of iterations.

Elements	Load P			Load M		
	Mesh (a)	Mesh (b)	Mesh (c)	Mesh (a)	Mesh (b)	Mesh (c)
Q4	0.093	0.034	0.027	0.093	0.031	0.022
QM6	0.993	0.632	0.051	1.000	0.727	0.045
P-S	0.993	0.798	0.221	1.000	0.852	0.167
PEAS7	0.982	0.795	0.217	-	-	-
QACM4	0.995	0.635	0.052	1.000	0.722	0.046
SU4	**0.993(0)**	**0.994(1)**	**0.994(1)**	**1.000(1)**	**1.000(1)**	**1.000(1)**
Reference	1.000 (the value: -0.1081)			1.000 (the value: -0.0054)		

Numerical examples

❖ Cantilever beam divided by two elements for mesh distortion test



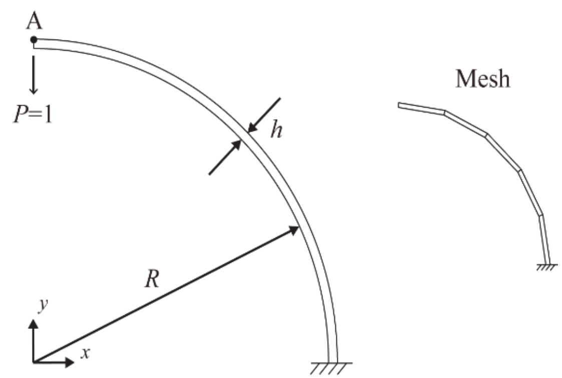
($E = 1500$, $\nu = 0.25$, thickness = 1.0)

Table. Normalized vertical displacements at point A and B in the cantilever beam for mesh distortion test with a distortion parameter *e*, data in bold are the results obtained by the elements proposed in this study, and the number in bracket is the number of iterations. (Normalized vertical displacements at point A/Normalized vertical displacements at point B)

Elements	*e*				
	1	2	3	4	4.9
Q4	0.144/0.141	0.098/0.097	0.083/0.083	0.071/0.072	0.62/0.062
QM6	0.673/0.627	0.624/0.544	0.657/0.536	0.669/0.512	0.648/0.468
P-S	0.679/0.629	0.631/0.550	0.672/0.548	0.700/0.630	0.703/0.498
SPS	1.100/1.000	1.208/1.000	1.332/1.000	1.479/1.000	1.639/0.999
SU4	**1.000/1.000(1)**	**1.000/1.000(1)**	**1.000/1.000(1)**	**1.000/1.000(1)**	**1.000/1.000(1)**
Exact	1.000/1.000 (the value: 100/100)				

Numerical examples

❖ Thin curving beam



((1) h/R=0.03 ($E = 365{,}010$, $\nu = 0$, thickness $= 1$) and

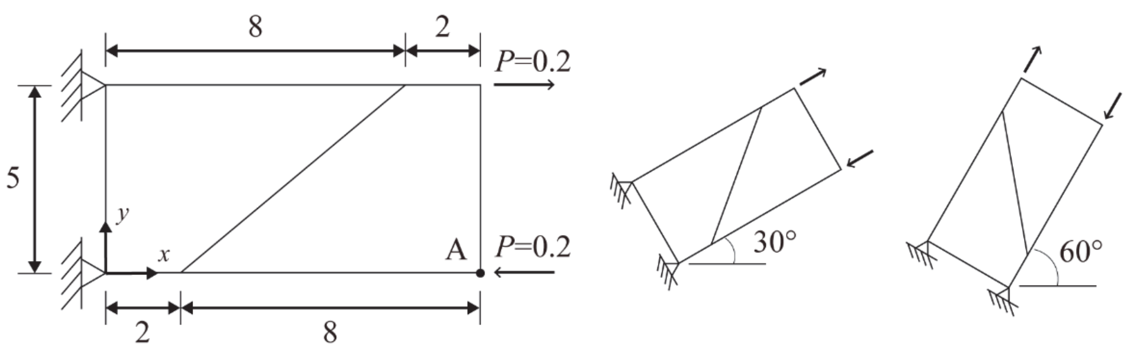
(2) h/R=0.006 ($E = 44{,}027{,}109$, $\nu = 0$, thickness $= 1$)).

Table. The vertical deflection at point A in the thin curving beam, data in bold are the results obtained by the elements proposed in this study, and the number in bracket is the number of iterations.

Elements	Tip deflection	
	h/R=0.03	h/R=0.006
Q4	0.016	0.001
QM6	0.650	0.173
QACM4	0.639	0.026
SU4	**1.005(7)**	**1.003(4)**
Exact	1.000	

Numerical examples

❖ Cantilever beam for rotation dependency test



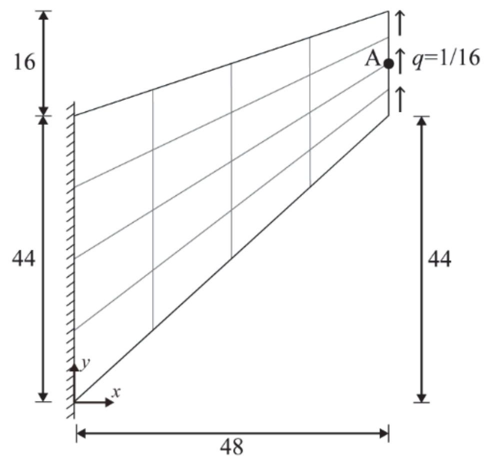
($E = 100$, $\nu = 0.3$, thickness = 1.0)

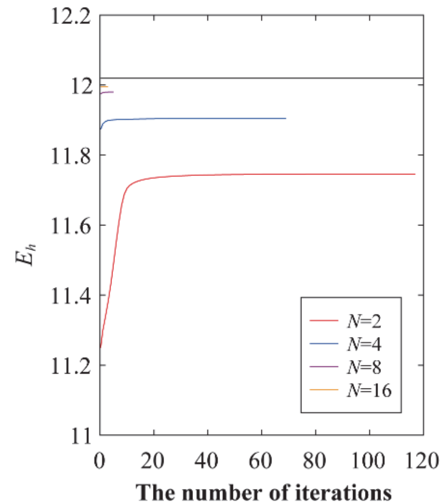
Table. The displacements at point A according to rotation angle in the cantilever beam for rotation dependency test using the SUFE, and the number in bracket at the last column is the number of iterations.

Rotation angle	Tip deflection u_A	v_A	$\sqrt{u_A^2 + v_A^2}$	Normalized
0°	2.4000E-02	4.8000E-02	0.05367	0.9938(1)
10°	1.5306E-02	5.1397E-02	0.05363	0.9931(4)
20°	6.1526E-03	5.3264E-02	0.05362	0.9929(5)
30°	3.1876E-03	5.3519E-02	0.05361	0.9928(6)
40°	1.2456E-02	5.2180E-02	0.05365	0.9934(6)
50°	2.1325E-02	4.9222E-02	0.05364	0.9934(8)
60°	2.9540E-02	4.4766E-02	0.05363	0.9932(6)
70°	3.6846E-02	3.8950E-02	0.05362	0.9929(5)
80°	4.3173E-02	-3.1982E-02	0.05373	0.9950(3)
90°	4.8000E-02	2.4000E-02	0.05367	0.9938(1)
Ref. solution	-	-	0.054	1.000

Numerical examples

❖ Cook's skew beam problem





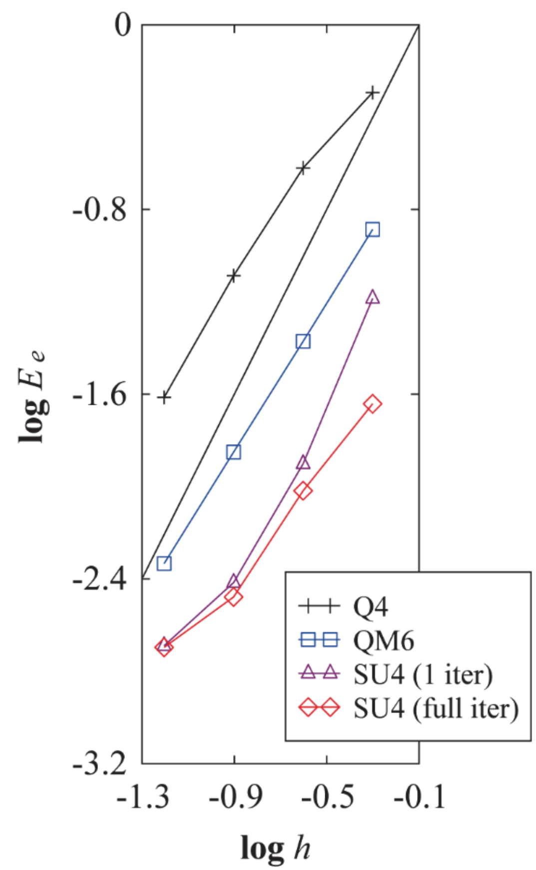


Table. The tip deflection (v_A) at point A according to mesh densities in Cook's skew beam problem, data in bold are the results obtained by the elements proposed in this study, and the number in bracket is the number of iterations.

Elements	v_A				Normalized			
	2×2	4×4	8×8	16×16	2×2	4×4	8×8	16×16
Q4	11.85	18.30	22.08	23.43	0.492	0.763	0.921	0.978
QM6	21.05	23.02	23.69	23.88	0.878	0.961	0.989	0.996
P-S	21.13	23.02	23.69	23.88	0.882	0.961	0.989	0.996
$\bar{\mathbf{B}}$-QE4	21.35	23.04	-	23.88	0.891	0.961	-	0.996
QACM4	20.74	22.99	23.69	-	0.865	0.959	0.989	-
GC-Q6	27.61	24.31	23.99	-	1.152	1.014	1.001	-
2D-MITC4/1	19.61	22.64	23.59	23.86	0.818	0.945	0.984	0.995
SU4	**23.80 (118)**	**23.93 (70)**	**23.96 (6)**	**23.95 (4)**	**0.993 (118)**	**0.998 (70)**	**1.000 (6)**	**0.999 (4)**
Ref. solution	23.9652				1.000			

Closure (Topic 2)

- Self-updated finite elements are proposed
  - New concepts and formulation are presented
    - Stiffness matrix updating method using displacements
    - Mode based element formulation

- The performance of the developed elements was evaluated through various numerical examples
  - SU4 showed promising ability in accuracy.

- Limitations of SUFE
  - Geometrical limit (out of train data distribution)
  - Iteration time

Ⅲ Conclusions & Future works

Conclusions

- ❖ These studies have great implications by showing that artificial intelligence can be used for finite element development.

- ❖ The proposed methods are not limited to the elements applied in this studies, and can be extended to various finite elements, including 3D solid, beam, and etc.

Future works

- The expansion of the research to other finite element.
  - Cost function design optimization
  - Network design optimization
  - Robust modeling research

- Nonlinear analysis.

- Model uncertainty research.

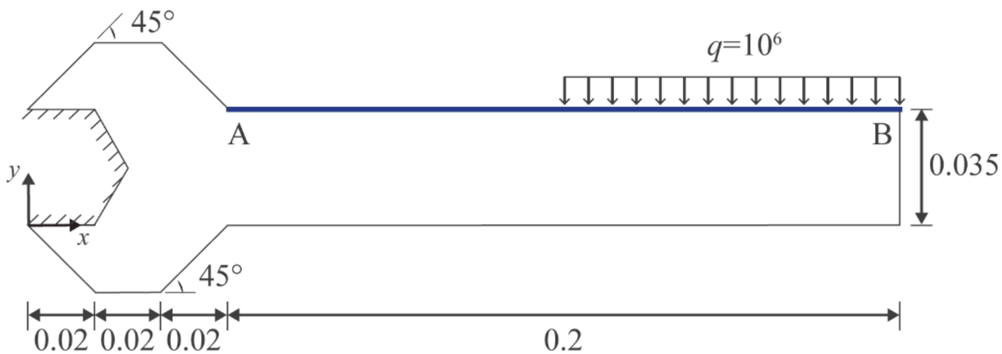
Thank you

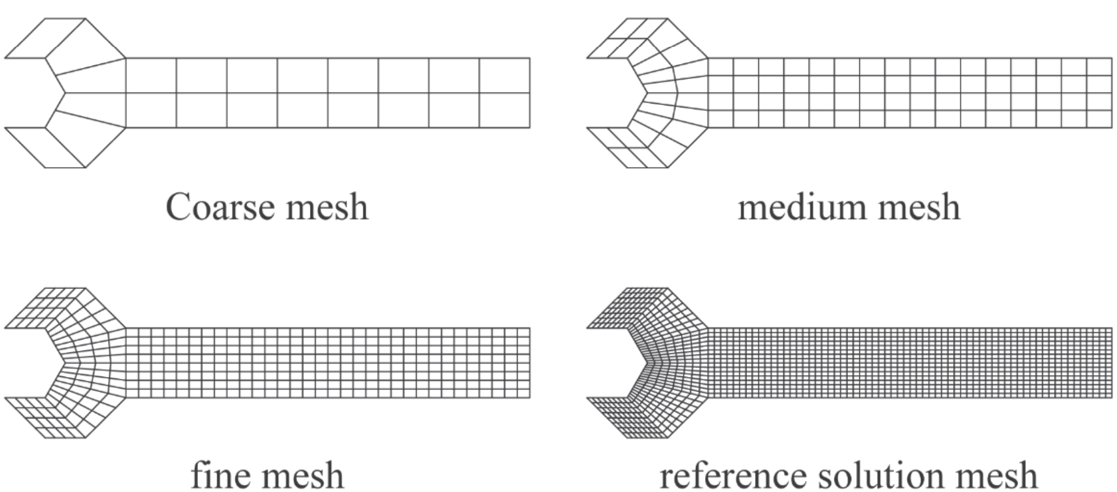
경청해 주셔서 감사합니다

Appendix A

❖ Mesh density of the reference data model

Problem description

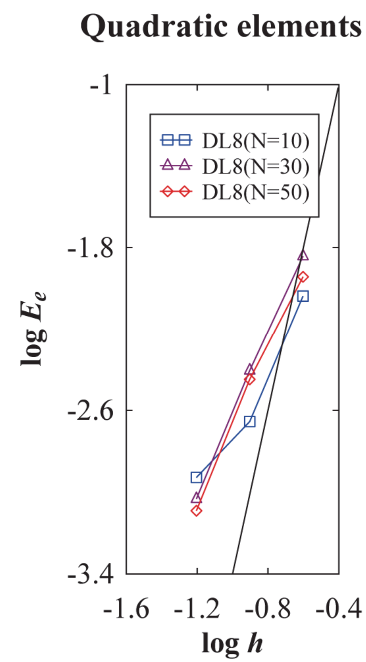


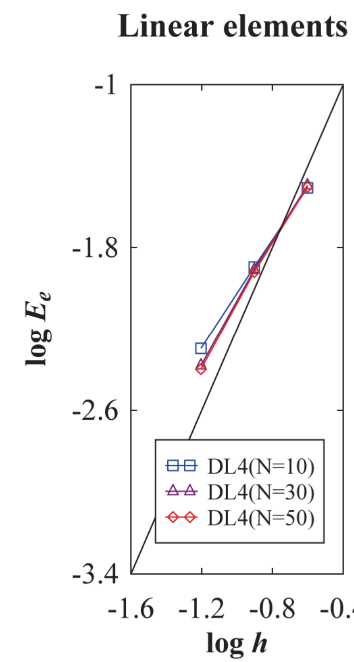


($E = 2.0\times10^{11}$, $\nu = 0.3$, thickness = 0.01)

Table. Averaged errors of the trained neural networks according to the mesh density of the reference data model (N).

N	Training data error (%)	Test data error (%)
10	1.57	2.68
30	1.55	1.98
50	1.24	1.67

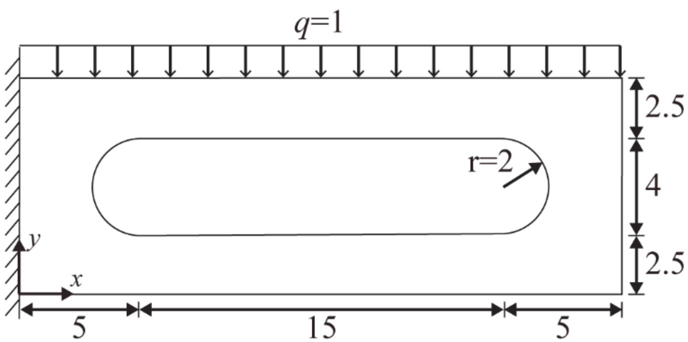


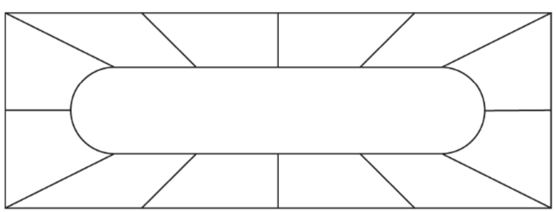


Appendix B

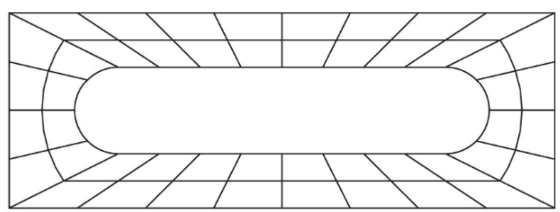
❖ Convergence behavior of the DL8 element in a curved geometry model

Problem description

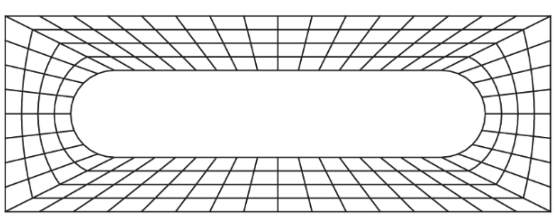




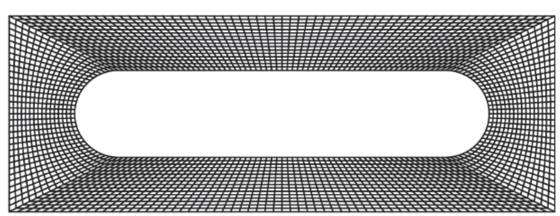
Coarse mesh (N = 2)



Medium mesh (N = 4)

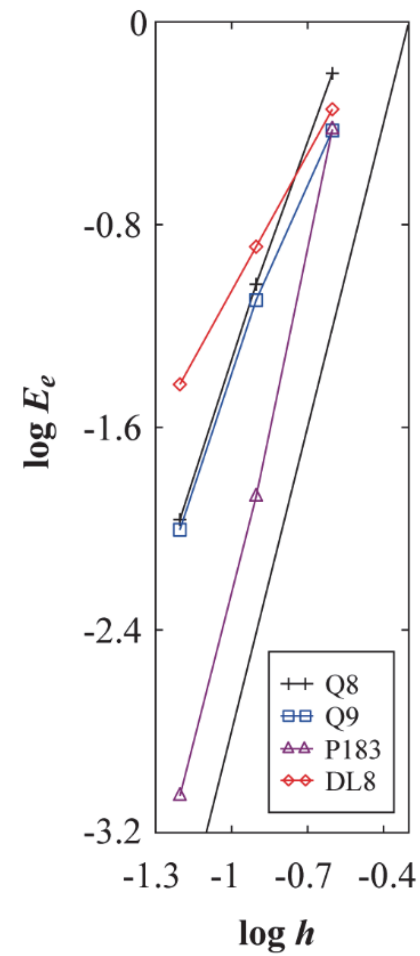


Fine mesh (N = 8)



Reference solution mesh

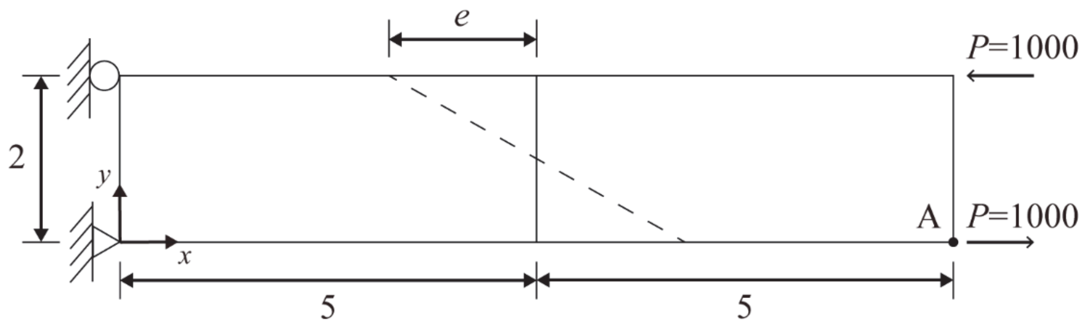
($E = 2.0\times10^{11}$, $\nu = 0.3$, thickness = 2.0)



Appendix C

❖ Application out of the trained distribution (extrapolation)

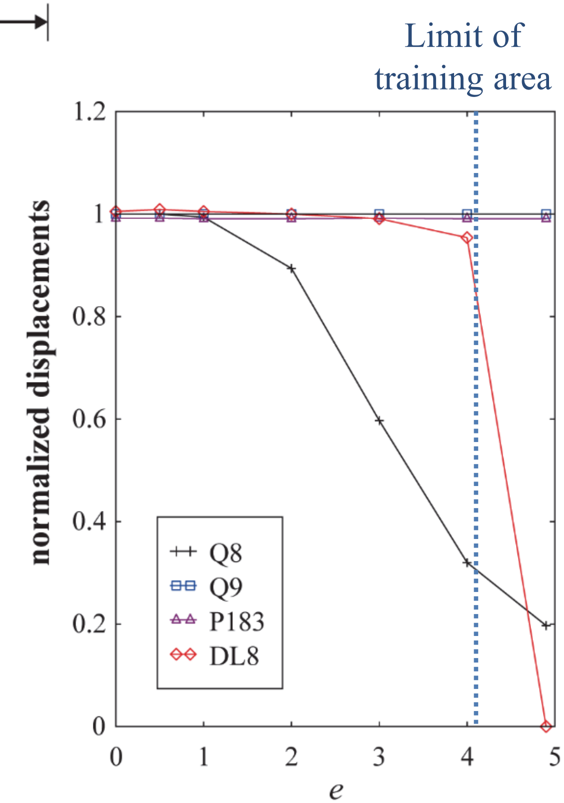
Problem description



($E = 1.5\times10^3$, $\nu = 0.25$, thickness = 1.0)

Table. Normalized vertical displacements at point A in the cantilever beam for mesh distortion test with a distortion parameter *e*

Elements	*e* (inside training area)						*e* (outside)
	0	0.5	1	2	3	4	4.9
Q8	1.000	1.000	0.994	0.894	0.597	0.320	0.197
Q9	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P183	0.992	0.992	0.991	0.991	0.992	0.991	0.991
DL8	**1.005**	**1.009**	**1.005**	**1.000**	**0.991**	**0.954**	**0.000**
Exact	1.000 (the value: 100)						



Appendix C

❖ How to overcome the extrapolation limits

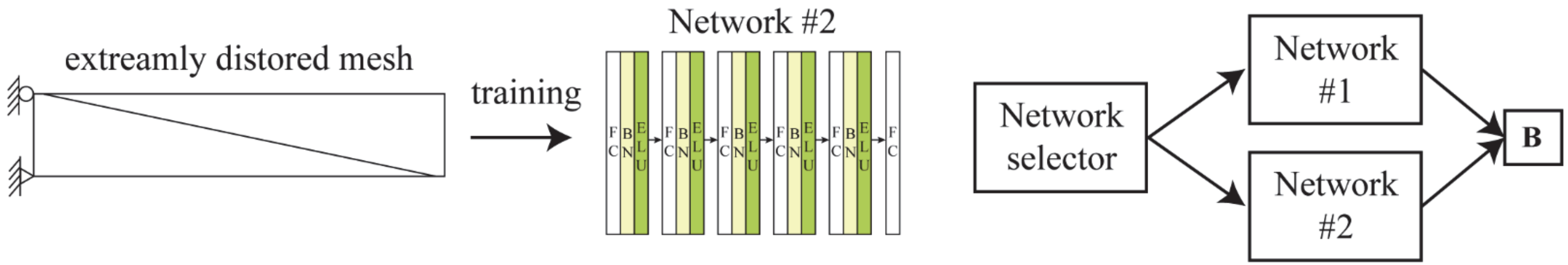
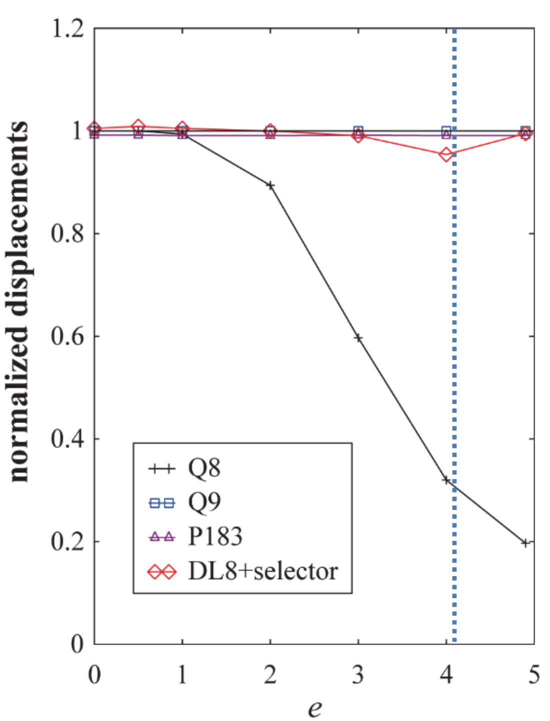


Table. **Normalized vertical displacements at point A in the cantilever beam for mesh distortion test with a distortion parameter *e***

Elements	*e* (inside training area)						*e* (outside)
	0	0.5	1	2	3	4	4.9
Q8	1.000	1.000	0.994	0.894	0.597	0.320	0.197
Q9	1.000	1.000	1.000	1.000	1.000	1.000	1.000
P183	0.992	0.992	0.991	0.991	0.992	0.991	0.991
DL8	1.005	1.009	1.005	1.000	0.991	0.954	0.000
DL8 +selector	**1.005**	**1.009**	**1.005**	**1.000**	**0.991**	**0.954**	**0.995**
Exact	1.000 (the value: 100)						

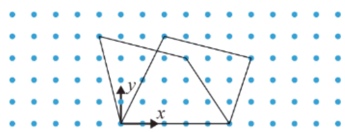


Appendix D

❖ Effect of data sampling method on network training

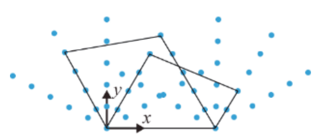
- Method 1

$$x_3 = -1 + i \cdot dx$$
$$y_3 = j \cdot dy$$
$$x_4 = -1 + k \cdot dx$$
$$y_4 = l \cdot dy \quad \text{for } i, j, k, l = 0, 1, 2, \ldots, p$$
$$dx = 3/p \text{ and } dy = 1/p$$



- Method 2

$$x_3 = 1 + i \cdot dr \cos(j \cdot d\theta)$$
$$y_3 = i \cdot dr \sin(j \cdot d\theta)$$
$$x_4 = k \cdot dr \cos(l \cdot d\theta)$$
$$y_4 = k \cdot dr \sin(l \cdot d\theta) \text{ for } i, j, k, l = 0, 1, 2, \ldots, p$$
$$dr = 1/p \text{ and } d\theta = 1/p$$



- Method 3

$$x_3 = 1 + \beta_3 \cos(\theta_3)$$
$$y_3 = \beta_3 \sin(\theta_3)$$
$$x_4 = \beta_4 \cos(\theta_4)$$
$$y_4 = \beta_4 \sin(\theta_4)$$

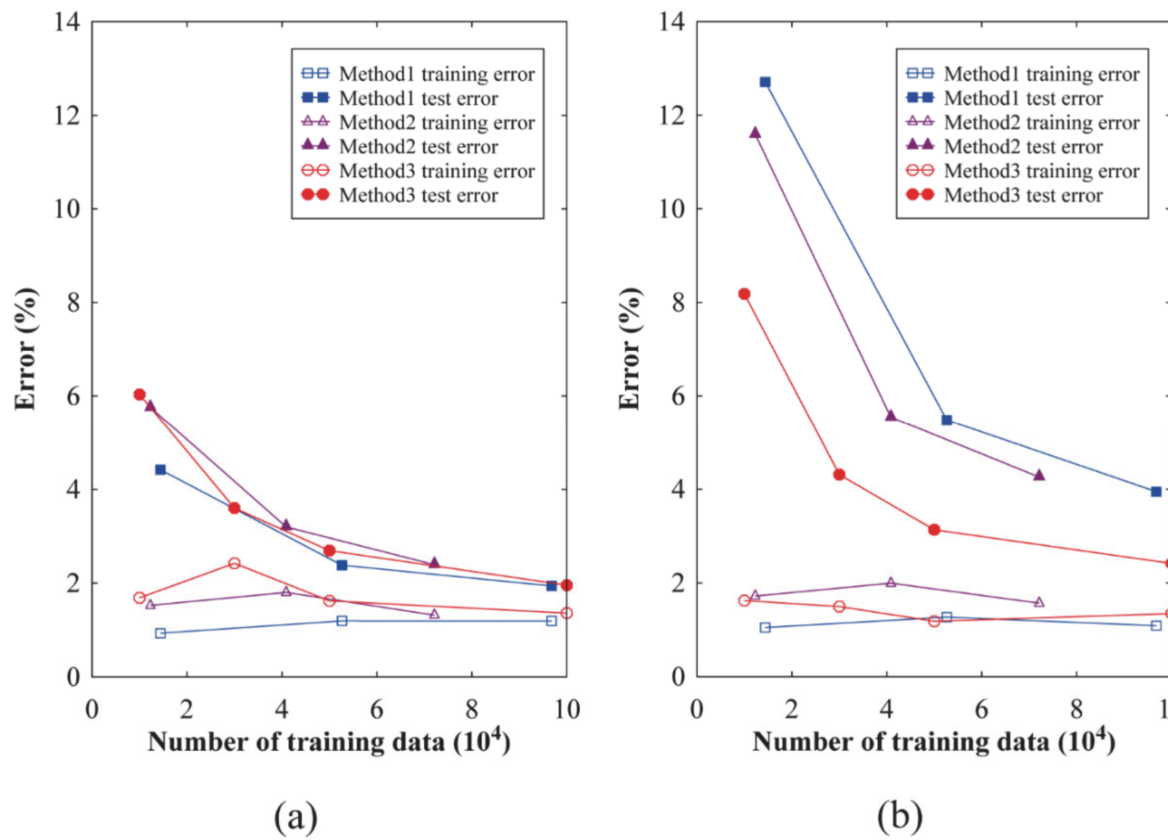


Figure. Error curves according to the data generation method using the test data generated in (a) Method 1 and (b) Method 3.

Appendix E

❖ Effect of input degree of freedom on the training

- In order to increase the input degree of freedom, we randomly rotate the normalized geometry

$$\mathbf{D}_x^{(n)} = \begin{bmatrix} \mathbf{x}_2^{(n)\mathrm{T}} & \mathbf{x}_3^{(n)\mathrm{T}} & \mathbf{x}_4^{(n)\mathrm{T}} \end{bmatrix} \rightarrow \text{network training}$$

Table. Averaged errors of training according to the depth and width of the network for each size of the increased input degree of freedom data (training data error (%)/test data error (%)).

Number of weights per layer	Number of Training data	Number of layers				
		4	5	6	7	8
378	300,000	4.46/5.96	4.31/6.02	4.20/6.08	4.22/6.21	4.06/6.16
378	600,000	4.20/4.97	4.04/4.94	4.02/5.01	4.03/5.13	3.88/5.08
756	300,000	3.04/4.86	3.07/5.04	2.72/4.93	2.90/5.54	2.81/6.00
756	600,000	2.64/3.64	2.72/3.88	2.85/4.15	2.73/4.06	2.75/4.15
1512	300,000	2.53/5.97	2.31/4.82	2.21/7.51	2.26/4.85	2.35/5.06
1512	600,000	2.25/3.51	2.32/3.74	2.02/3.59	2.15/4.12	2.17/3.66

Appendix F

- Effect of network structure and data size
  - The number of epochs : 10,000

Table. Averaged errors of training according to the depth and width of the network in Fig. 4.4 for each size of training data (training data error (%)/test data error (%)).

Number of weights per layer (network width)	Number of Training data	Number of layers (network depth)				
		2	4	6	8	10
189	10,000	11.70/22.19	4.49/11.23	3.17/10.32	3.15/10.22	3.16/11.14
189	50,000	11.55/14.02	3.65/5.69	2.98/5.11	2.82/5.11	2.71/4.96
189	100,000	11.74/13.20	3.38/4.48	2.80/4.16	2.66/3.97	2.67/4.10
378	10,000	11.08/20.77	3.77/10.55	2.67/9.73	2.44/9.37	2.50/11.30
378	50,000	10.28/12.86	2.57/4.63	1.89/4.04	1.97/4.22	1.97/4.39
378	100,000	9.99/11.51	2.52/3.81	2.05/3.32	1.78/3.13	1.70/3.08
756	10,000	9.54/19.58	2.60/9.06	2.14/8.83	3.04/9.61	2.17/8.46
756	50,000	8.96/11.65	1.92/3.93	1.83/3.95	1.46/3.64	1.67/3.82
756	100,000	9.04/10.47	1.86/3.08	1.57/2.83	1.40/2.63	1.50/2.86

Appendix G

- Effect of network hyper parameter
  - The number of training data : 50,000
  - The number of test data : 10,000
  - The number of epochs : 10,000

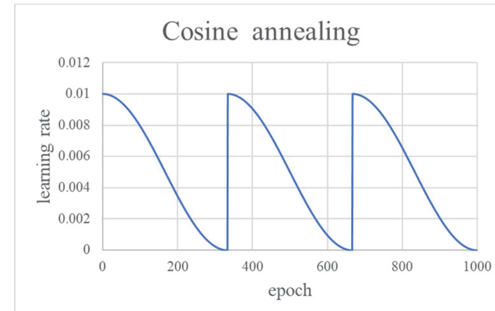


Table. Averaged errors of training according to the learning rate, batch size, activation function type (training data error (%)/test data error (%)).

Type of activation function	Batch size	The learning rate					
		0.001	**0.005**	**0.01**	**0.05**	**0.01→0 linear**	**0.01→0 cosine**
ELU	10,000	4.27/6.39	4.23/6.29	4.32/6.19	7.29/10.52	2.43/4.52	2.35/4.33
ELU	25,000	5.60/7.51	4.63/6.57	4.42/6.21	7.40/10.37	2.20/4.30	2.24/4.44
ELU	50,000	9.68/11.72	7.50/9.41	8.38/10.67	10.76/12.79	2.55/4.81	2.62/4.82
Sigmoid	10,000	7.86/9.79	10.90/13.40	14.51/17.82	30.42/37.42	2.07/3.96	2.53/4.90
Sigmoid	25,000	9.47/11.43	8.87/11.25	13.06/15.27	22.17/26.39	2.41/4.35	2.68/4.76
Sigmoid	50,000	16.94/18.80	12.03/13.59	16.50/20.07	24.41/30.25	3.33/5.19	3.39/5.32
tanh	10,000	6.66/8.97	7.37/10.00	8.55/11.82	12.38/19.19	2.20/4.53	2.70/5.01
tanh	25,000	8.00/10.63	7.41/8.89	8.13/10.21	13.13/16.06	1.97/4.21	2.87/5.46
tanh	50,000	13.26/15.70	9.84/12.45	11.02/13.28	20.43/24.50	2.60/4.90	2.68/5.08
Leaky ReLU	10,000	4.40/6.83	6.16/8.24	4.91/7.25	6.91/9.55	2.50/4.95	2.81/5.36
Leaky ReLU	25,000	6.35/9.47	4.08/6.97	4.01/6.45	7.27/10.24	2.18/5.19	2.40/5.27
Leaky ReLU	50,000	9.57/13.34	6.73/9.10	6.76/8.79	9.23/12.33	2.37/5.77	2.84/6.00

Appendix H

- Effect of optimizer
  - The number of training data : 50,000
  - The number of test data : 10,000
  - The number of epochs : 10,000

Table. Averaged errors of training according to the optimizer (training data error (%)/test data error (%)).

Type of optimizer	Batch size	The learning rate 0.001	0.005	0.01	0.05	0.01→0 linear	0.01→0 cosine
SGD	10,000	89.35/95.12	50.74/54.37	34.74/37.67	13.87/16.72	49.03/52.63	49.40/53.01
SGD	25,000	116.3/123.6	76.85/81.66	59.29/62.98	24.56/27.12	71.72/76.04	71.84/76.22
SGD	50,000	127.7/134.7	94.81/100.6	78.46/82.64	38.06/41.14	88.93/93.96	89.82/94.78
Adagrad	10,000	85.41/92.67	46.64/50.79	28.60/31.65	9.29/11.33	41.02/44.46	42.12/45.78
Adagrad	25,000	99.44/106.6	64.29/68.68	45.70/49.20	15.82/18.31	57.77/61.76	59.26/63.11
Adagrad	50,000	114.4/122.1	78.12/83.10	60.97/64.78	25.36/27.79	71.30/75.63	72.42/76.69
Adam	10,000	4.27/6.39	4.23/6.29	4.32/6.19	7.29/10.52	2.43/4.52	2.35/4.33
Adam	25,000	5.60/7.51	4.63/6.57	4.42/6.21	7.40/10.37	2.20/4.30	2.24/4.44
Adam	50,000	9.68/11.72	7.50/9.41	8.38/10.67	10.76/12.79	2.55/4.81	2.62/4.82
Adam+SGD	10,000	4.38/6.26	3.74/5.81	4.92/6.88	8.21/11.34	2.29/4.40	2.52/4.44
Adam+SGD	25,000	6.42/8.40	5.01/6.87	5.41/7.31	8.35/11.03	2.03/4.31	2.62/4.75
Adam+SGD	50,000	11.08/13.28	7.04/8.99	8.58/10.79	10.10/12.71	2.96/5.21	3.23/5.47